NASA/CR—2003-212366 MTI 94TM7

Source Listings for Computer Code KTK
Labyrinth Seal Design Model

Mechanical Technology, Inc.
Latham, New York

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

Export Administration Regulations (EAR) Notice

This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.

October 2005

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2003-212366 MTI 94TM7

Source Listings for Computer Code KTK
Labyrinth Seal Design Model

Mechanical Technology, Inc.
Latham, New York

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

Export Administration Regulations (EAR) Notice

This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.

Prepared under Contract NAS3–25644

National Aeronautics and
Space Administration

Glenn Research Center

October 2005

Document History

The source and executable files for the CFD Seal Analysis Industrial Codes, of which this is a part, were released as LEW–16582 in 1998. This report was originally published by Mechanical Technology, Inc., in July 1994.

Document Availability Change Notice

NASA/CR—2003-212366

Source Listings for Computer Code KTK Labyrinth Seal Design Model

Mechanical Technology, Inc.

This document was published in July 2003 with the restriction Export Administration Regulations (EAR). It was changed to Unclassified/Unlimited July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

Per the STI Program Office and Code I at Headquarters, you may modify copies in your possession. The restriction notice on the cover, title page, and Report Documentation Page should be boldly crossed out and the above statement printed clearly above or below it.

This work was sponsored by the Low Emissions Alternative Power Project of the Vehicle Systems Program at the NASA Glenn Research Center.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

Program Listing

```
C -----------------------------------------------------------------
C     File:    KTK.FOR
C     Program: KTK
C     Version: 2
C
C     Revision History:
C     (1) Modified December 1993 to compile under OS/2 using the
C     WATCOM F77/386 V9.0 compiler.
C     (2) Modified June 1994 to change the input formats for
C     integration with the CFD user interface shell.
C -----------------------------------------------------------------
      PROGRAM KTK
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL D1
      CHARACTER*5   DIRECT, ALTSD, STLD
      CHARACTER*5   TYPE, STEP, STRGHT, TMIX, XTYPE
      CHARACTER*5   AOPT, AOPT1, AOPTD
      CHARACTER*5   OPTM, OPTIM
      CHARACTER*8   TYP(3)
      CHARACTER*4   TITLE, TITLEC
      CHARACTER*80  CH30, FILN(5)
C
      COMMON /C12R/ WVAL(30),TT(100),AA(100)
      COMMON /C12I/ NV,NQ
      COMMON /C13R/ PARVAL,AMIN,ROGCN
      COMMON /C13RA/TITLE(20),TITLEC(20),AOPTD
C
      COMMON /C13I/ MG,NOSETS,ISETS
      COMMON /C14R/ G,GP1,GP1GM1,GP1OG,GP1O2,GM1O2,XK(100)
      COMMON /C15R/ C1,GP1GM2,CMAX,GM1O2D,GOGM1
      COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
     1,XKPRT(100),XKPRTP(100)
      COMMON /C124I/ KURVE(100),IDPORQ(100)
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /C1234I/ NL1,KODE
      COMMON /C78R/ XCL,XKP,XKH,XKT,XSH,XDTC,XDIRCT,XKR,XKTH
     1,XVGDIA,XELDIA,XLENGH,XKBETA
      COMMON /C78RA/ XTYPE
      COMMON /OPTDAT/ ALENGH,DMIN,DMAX,ALPHA,PRATIO,XMW,ARFACT,NOW
     1,ICURVE,ICVTYP,IPRINT
      COMMON /OPTX/ AOPT
      COMMON /VECTR/ CL(30),AKR(30),AKT(30),AKP(30),AKH(30),ASH(30),ADTC
     1(30),AKTH(30),AKBETA(30),DIA(30),ROUGH(30),TR(30),A4FLOD(30),DELC
     2(30),DELE(30),KCCO(30),KECO(30),PKR(30)
      COMMON /CROUGH/ ROUFLD(100),ROUFED(100),ROUFDH(100)
      COMMON /CMIX/ TMIX

C
      DIMENSION ALPH(30), XMUL(30), TANAL(30), OOTANA(30), VLIM(3,2)
C
      DATA UNVGAS/1545.3/,GC/32.174/,P0/14.7/
      DATA PIE/3.14159265/
      DATA ALTSD/'LTSD '/,STLD/'STLD '/
      DATA STEP/'STEP '/,STRGHT/'STRAI'/
      DATA AOPT1/'UNITY'/
      DATA TYP/'STRAIGHT',' STEPPED','   MIXED'/
      DATA OPTM/'OPTIM'/
      DATA VLIM/0.21,5.1,1.76,0.50,5.1,1.90/
C
C
C
      BTAN(TDUM) = TAN(TDUM*RADIAN)
```

```
      BSIN(CDUM) = SIN(CDUM*RADIAN)
      BSQRT(SDUM)= SQRT(SDUM)
      BABS(ADUM) = ABS(ADUM)
      BMAX1(ARG1,ARG2) = DMAX1(ARG1,ARG2)
C
C*********************************************************************
C.....SET FILE NAMES
C
C     INPUT FILE NAME SET TO "KTK.INP"
C
      FILN(1) = 'KTK.INP'
      OPEN(5,FILE=FILN(1),STATUS='OLD')
C
C*********************************************************************
C
C     START  READING INPUT DATA
C
      RADIAN=PIE/180.
      XMW=28.97
      G=1.4
C
10    CONTINUE
C
C ******************************************************************
C RECORD 1 FORMAT WAS CHANGED TO ALLOW LARGER WIDTH FOR FLOATING
C POINT NUMBERS
C RECORD 1B WAS ADDED TO INPUT OUTPUT FILE NAME
C ******************************************************************
      READ(5,20,END=760) IPRINT,ICURVE,ICVTYP,NOW,NOSETS,GG,XM,PZ
20    FORMAT (5I5,3E12.4E3)
C
      READ(5,21,END=760) CH30
21    FORMAT (A80)
C
C
C ******************************************************************
C
C     SET OUTPUT FILE NAMES
C
      CALL FILES(CH30,79,FILN)
      OPEN(6, FILE=FILN(2), STATUS='UNKNOWN')
      OPEN(7, FILE=FILN(3), STATUS='UNKNOWN')
C
      IF (XM.GT.0.) XMW=XM
      IF (PZ.GT.0.) P0=PZ
      IF (GG.GT.0.) G=GG
      IF (G.LT.2.) GO TO 40
      TRG=G
      CALL GAMCAL (TRG,P0,G)
      WRITE (6,30) TRG,P0,G
30    FORMAT (' TEMP',G14.6,'P0',G14.6,'CALCULATED GAMMA',G14.6)
C
C     **   SET INITIAL VARIABLES WHICH ARE FUNCTIONS OF G (GAMMA)  *****
C
40    ROGCN=UNVGAS/(XMW*GC)
      C1=BSQRT(G/ROGCN)
      GM1=G-1.
      GP1=G+1.
      GP1GM2=GP1/(2.*GM1)
      GM1O2=GM1/2.
      GM1O2D=GM1O2
      CMAX=C1/(1.+GM1O2)**GP1GM2
      GOGM1=G/GM1
```

```
      GP1GM1=GP1/GM1
      GP1OG=GP1/G
      GP1O2=GP1/2.
      ROGCN=UNVGAS/(XMW*GC)
      ISETS=0
C
50    CONTINUE
C
C *****************************************************************
C RECORDS 2 AND 3 REMAIN UNCHANGED
C *****************************************************************
      READ(5,60,END=760) TITLE
C
      READ(5,60) TITLEC
60    FORMAT (20A4)
C *****************************************************************
C
70    CONTINUE
      PARVAL=0.0
C
C *****************************************************************
C RECORD 4 WAS SPLIT IN TWO TO ALLOW FOR LARGER WIDTH FOR
C FLOATING POINT NUMBERS:
C
C       RECORD 4A: NOKNIF,NOKNRC,TYPE,DIRECT,AOPT,OPTIM
C       RECORD 4B: PT(1),ALENGH,DMIN,DMAX,ARFACT,ALPHA,PRAT
C *****************************************************************
      READ(5,80,END=760) NOKNIF,NOKNRC,TYPE,DIRECT,AOPT,OPTIM
80    FORMAT (2I5,4A5)
C
      READ(5,81,END=760) PT(1),ALENGH,DMIN,DMAX,ARFACT,ALPHA,PRAT
81    FORMAT (4E12.4E3,3F7.2)
C *****************************************************************
C
      PRATIO=PRAT
      IF (PRAT.GT.0..AND.PRAT.LT.1.) PRATIO=1./PRAT
C
C                   CHECK ON SEAL TYPE READ
C
      IF (TYPE.EQ.STEP.OR.TYPE.EQ.STRGHT.OR.TYPE.EQ.TMIX) GO TO 100
      WRITE (6,90) TYPE,STEP,STRGHT,TMIX
90    FORMAT (' INPUT SEAL TYPE ----',A5,'---- NOT RECOGNIZED'/' PERMISS
     1IBLE TYPES ARE',3(3X,A5))
      STOP
C
C                   CHECK TO BE SURE NOKNRC = NOKNIF  FOR TYPE = MIXED
C
100   IF (TYPE.NE.TMIX.OR.NOKNRC.EQ.NOKNIF) GO TO 120
      WRITE (6,110) NOKNRC,NOKNIF
110   FORMAT (' INPUT FOR MIXED SEALS MUST HAVE NUMBER OF KNIFE RECORDS
     1 READ (NOKNRC) EQUAL TO NUMBER OF KNIVES (NOKNIF)'/' IN THIS DATA
     2SET  NOKNIF =',I5,'    NOKRC =',I5)
      STOP
120   CONTINUE
      AOPTD=AOPT
      IF (NOKNRC.LE.0) NOKNRC=1
      IF (ARFACT.LE.0.0) ARFACT=1.0
C
C        INITIALIZE KNIFE GEOMETRY PARAMETERS
C
      DO 130 I=1,NOKNIF
      CL(I)=0.0
      AKR(I)=0.0
```

```
      AKT(I)=0.0
      AKP(I)=0.0
      AKH(I)=0.0
      ASH(I)=0.0
      ADTC(I)=0.0
      AKTH(I)=0.0
      AKBETA(I)=0.0
      TR(I)=0.0
      DIA(I)=0.0
      ROUGH(I)=0.0
      A4FLOD(I)=0.0
      XMUL(I)=1.0
      ALPH(I)=0.0
      KCCO(I)=0
      KECO(I)=0
130   CONTINUE
C
C ***************************************************************
C RECORD 5 WAS SPLIT IN THREE TO ALLOW FOR LARGER WIDTH FOR
C FLOATING POINT NUMBERS:
C
C      RECORD 5A: CL(I),AKR(I),AKT(I),AKP(I),AKH(I),ASH(I)
C      RECORD 5B: ADTC(I),AKTH(I),AKBETA(I),DIA(I),ROUGH(I),TR(I)
C      RECORD 5C: KCCO(I),KECO(I),A4FLOD(I)
C ***************************************************************
      DO 145 I = 1, NOKNRC
      IF(NOSETS .GE. 0) READ(5,140) CL(I),AKR(I),AKT(I),AKP(I),
     1                              AKH(I),ASH(I)
140   FORMAT (6E12.4E3)
C
      IF(NOSETS .GE. 0) READ(5,141) ADTC(I),AKTH(I),AKBETA(I),
     1                              DIA(I),ROUGH(I),TR(I)
141   FORMAT (E12.4E3, 2F7.3, 2E12.4E3, F7.3)
C
      IF(NOSETS .GE. 0) READ(5,142) KCCO(I),KECO(I),A4FLOD(I)
142   FORMAT (2I5, F7.3)
145   CONTINUE
C ***************************************************************
C
      ISETS=ISETS+1
C
C ***************************************************************
C RECORD 6 FORMAT WAS CHANGED TO ALLOW FOR LARGER WIDTH FOR
C FLOATING POINT NUMBERS:
C ***************************************************************
      IF(NOW .GT. 0) READ(5,150) (WVAL(I), I = 1, NOW)
150   FORMAT (6E12.4E3)
C ***************************************************************
C
C        FINISHED READING INPUT DATA
C
      IF (OPTIM.EQ.OPTM) GO TO 730
      IF (NOSETS.LT.0) GO TO 160
      WRITE (6,580)
      WRITE (6,590)
      WRITE (6,600) TITLE,TITLEC
C
C        SET UP GEOMETRY DATA WHERE NOT READ IN
C
160   IF (TR(1).LE.0.0) TR(1)=530.
      IF (TYPE.NE.TMIX) GO TO 240
C
C             MIXED SEALS    -----  MAKING SURE DIAMETERS ARE READ IN
```

```
C                                       PROPERLY
C
      IF (DIA(1).GT.0.) GO TO 180
      WRITE (6,170)
170   FORMAT (' INITIAL DIAMETER MUST BE READ IN FOR A MIXED TYPE SEAL')
      STOP
180   IF (DIA(2).LE.0.) GO TO 220
      IF (NOKNIF.LT.3) GO TO 270
      DO 190 I=3,NOKNIF
      IF (DIA(I).LE.0.) GO TO 200
190   CONTINUE
      GO TO 270
200   WRITE (6,210)
210   FORMAT (' IF DIA(2) IS READ IN FOR A MIXED TYPE SEAL, ALL DIAMETER
     1S MUST BE READ IN')
      STOP
C
C            PROGRAM STOPPED ABOVE IF DIAMETERS ARE NOT READ CORRECTLY
C
C
220   DO 230 I=2,NOKNIF
      DIA(I)=DIA(I-1)+2.*ASH(I)
      IF (NOSETS.GE.0.AND.ALENGH.LE.0.) DIA(I)=DIA(I)-2.*(CL(I)-CL(I-1))
230   CONTINUE
      IF (KECO(1).NE.0) GO TO 270
      KECO(1)=1
      IF (BABS(DIA(1)-DIA(2)).GT.CL(1)) KECO(1)=-1
      GO TO 270
C
C             STRAIGHT AND STEPPED SEALS
C
240   IF (NOSETS.GE.0) AVGDIA=BMAX1(DMIN,DMAX)
      IF (DIA(1).LE.0.0) DIA(1)=AVGDIA
      IF (KCCO(1).NE.0) GO TO 250
      KCCO(1)=-1
      IF (TYPE.EQ.STRGHT) KCCO(1)=1
250   IF (KECO(1).NE.0) GO TO 260
      KECO(1)=-1
      IF (TYPE.EQ.STRGHT) KECO(1)=1
260   IF (NOKNIF.LE.1) GO TO 390
270   SUM=DIA(1)
C
C        ALL SEAL TYPES
C
      DO 330 I=2,NOKNIF
      IF (TYPE.EQ.TMIX) GO TO 280
      IF (CL(I).LE.0.0) CL(I)=CL(1)
      IF (AKR(I).LE.0.0) AKR(I)=AKR(1)
      IF (AKT(I).LE.0.0) AKT(I)=AKT(1)
      IF (AKP(I).LE.0.0) AKP(I)=AKP(1)
      IF (AKH(I).LE.0.0) AKH(I)=AKH(1)
      IF (ASH(I).LE.0.0) ASH(I)=ASH(1)
      IF (ADTC(I).LE.0.0) ADTC(I)=ADTC(1)
      IF (AKTH(I).LE.0.0) AKTH(I)=AKTH(1)
      IF (AKBETA(I).LE.0.0) AKBETA(I)=AKBETA(1)
      IF (TR(I).LE.0.0) TR(I)=TR(1)
      IF (ROUGH(I).LE.0.0) ROUGH(I)=ROUGH(1)
      IF (A4FLOD(I).LE.0.0) A4FLOD(I)=A4FLOD(1)
      IF (TYPE.NE.TMIX) GO TO 300
C
C             NOTE:  ABOVE STATEMENT WILL ALWAYS BE TRUE
C
C             CHECK FOR MIXED CASE TO SEE IF CARRY OVER TO BE ASSESSED
```

```
C
C                    NO CARRYOVER IF DEL DIA > CL
C
280   IF (KCCO(I).NE.0) GO TO 290
      KCCO(I)=1
      IF (BABS(DIA(I)-DIA(I-1)).GT.CL(I)) KCCO(I)=-1
290   IF (KECO(I).NE.0.OR.I.EQ.NOKNIF) GO TO 310
      KECO(I)=1
      IF (BABS(DIA(I)-DIA(I+1)).GT.CL(I)) KECO(I)=-1
      GO TO 310
C
C             NON MIXED CASES
C
300   IF (KCCO(I).EQ.0) KCCO(I)=KCCO(1)
      IF (KECO(I).EQ.0) KECO(I)=KECO(1)
310   IF (DIA(I).GT.0.0) GO TO 320
      IF (TYPE.EQ.STRGHT) DIA(I)=DIA(I-1)
320   SUM=SUM+DIA(I)
330   CONTINUE
      IF (TYPE.NE.STEP) GO TO 380
C
C             STEP SEALS
C
C             DETERMINE DIAMETERS IF DMAX OR DMIN IS READ
C
C             D1 = TRUE  ===> DIA(     1) GIVEN
C             D1 = FALSE ===> DIA(NOKNIF) GIVEN
C
      D1=.TRUE.
      IF (DMAX.GT.0.) GO TO 340
      IF (DMIN.LE.0.) GO TO 380
      IF (DIRECT.EQ.STLD) DIA(1)=DMIN
      IF (DIRECT.EQ.ALTSD) DIA(NOKNIF)=DMIN
      IF (DIRECT.EQ.ALTSD) D1=.FALSE.
      GO TO 350
340   IF (DIRECT.EQ.ALTSD) DIA(1)=DMAX
      IF (DIRECT.EQ.STLD) DIA(NOKNIF)=DMAX
      IF (DIRECT.EQ.STLD) D1=.FALSE.
350   SUM=DIA(1)
      IF (.NOT.D1) SUM=DIA(NOKNIF)
      DO 370 I=2,NOKNIF
      IF (.NOT.D1) GO TO 360
      IF (DIRECT.EQ.STLD) DIA(I)=DIA(I-1)+2.*ASH(I)
      IF (DIRECT.EQ.ALTSD) DIA(I)=DIA(I-1)-2.*ASH(I)
      IF (NOSETS.GE.0.AND.ALENGH.LE.0.) DIA(I)=DIA(I)-2.*(CL(I)-CL(I-1))
      SUM=SUM+DIA(I)
      GO TO 370
360   J=NOKNIF-I+1
      IF (DIRECT.EQ.STLD) DIA(J)=DIA(J+1)-2.*ASH(J+1)
      IF (DIRECT.EQ.ALTSD) DIA(J)=DIA(J+1)+2.*ASH(J+1)
      IF (NOSETS.GE.0.AND.ALENGH.LE.0.) DIA(J)=DIA(J)-2.*(CL(J+1)-CL(J))
      SUM=SUM+DIA(J)
370   CONTINUE
C
C             ALL SEAL TYPES
C
C             INITIALIZE 1ST KNIFE UPSTREAM VARIABLES IF ONLY ONE KNIFE
C                 RECORD IS READ, I.E. NOKNRC = 1
C
380   IF (TYPE.EQ.STEP.AND.NOKNRC.EQ.1) ASH(1)=0.
      IF (TYPE.EQ.STEP.AND.NOKNRC.EQ.1) AKP(1)=0.
      IF (TYPE.EQ.STEP.AND.NOKNRC.EQ.1) ADTC(1)=0.
      DUM=NOKNIF
```

```
      AVGDIA=SUM/DUM
390   KCCO(1)=-1
      KECO(NOKNIF)=-1
C
C              CHECK RANGES FOR NOKNRC > 1 (RECORD READ FOR EACH KNIFE)
C
      IF (NOKNRC.LE.1) GO TO 400
      CALL INCHEK (CL,AKT,AKP,AKH,ASH,ADTC,AKTH,ROUGH,TYPE,NOKNRC,IERR
     1,NOKNIF,DIRECT)
400   CONTINUE
C
C        SET UP RESTRICTION DATA FOR EACH KNIFE
C
      NL=-1
C
C     *****************     START LOOP OVER EACH KNIFE    *************
C
C                 THREE RESTRICTIONS SET UP FOR EACH KNIFE
C
      DO 500 I=1,NOKNIF
      IP1=I+1
      IM1=I-1
      NL=NL+3
      NLP1=NL+1
      NLP2=NL+2
C
C              CALCULATE ALPHA FOR STRAIGHT SEALS OR XMUL FOR STEP SEALS
C
      XTYPE=TYPE
      ALPH(IP1)=ALPHA
      IF (PARVAL.GT.0..AND.NOSETS.LT.0) XMUL(I)=PARVAL
C
      IF (NOSETS.LT.0) GO TO 450
      XCL=CL(I)
      XKT=AKT(I)
      XKP=AKP(I)
      XKH=AKH(I)
      IF (TYPE.EQ.STRGHT) GO TO 440
C
C             BEGIN CALCULATIONS FOR STEPPED OR MIXED SEALS ONLY
C
      XDTC=ADTC(I)
      XSH=BABS(ASH(I))
      XKL=XKP-XKT
      RDTC=XDTC/XCL
      RKT=XKT/XCL
      RKH=XKH/XCL
      RSH=XSH/XCL
      RDTCKL=XDTC/XKL
      RKLKH=XKL/XKH
C
C              RANGE CHECK WITH PARAMETER VALUE ADJUSTED IF OUT OF RANGE
C                   AND PARAMETER CAN NOT BE EXTRAPOLATED
C
      JD=1
      IF (DIRECT.EQ.ALTSD) JD=2
      IF (TYPE.EQ.TMIX.AND.ASH(I).LT.0.) JD=2
      IF (RKT.GE.VLIM(1,JD)) GO TO 410
      XKT=XCL*VLIM(1,JD)
      RKT=XKT/XCL
      XKL=XKP-XKT
      RKLKH=XKL/XKH
      RDTCKL=XDTC/XKL
```

```
410   IF (RKH.GE.VLIM(2,JD)) GO TO 420
      XKH=XCL*VLIM(2,JD)
      RKH=XKH/XCL
420   IF (RKLKH.LE.VLIM(3,JD)) GO TO 430
      TKH=XKL/VLIM(3,JD)
      XKH=BMAX1(XKH,TKH)
      RKH=XKH/XCL
      RKLKH=XKL/XKH
430   CONTINUE
C
C             CALCULATION OF XMUL FOR STEPPED AND MIXED SEALS
C
      XMUL(I)=1.
      IF (XSH.GT.XCL) XMUL(I)=1.168*RDTC/BSQRT(RDTC**2+0.500)*RKT**0.156
     1*RDTCKL**0.109/RKH**0.0075*RKLKH**0.095/RSH**0.025
      IF (TYPE.EQ.STEP.AND.DIRECT.EQ.STLD) GO TO 440
      IF (TYPE.EQ.TMIX.AND.(ASH(I)+CL(I)).GT.0.) GO TO 440
C
C         CORRECTION ON XMUL FOR LTSD DIRECTION
C
      DUM=1.0
      IF (XSH.GT.XCL) DUM=0.89*RKH**0.061
      XMUL(I)=XMUL(I)*DUM
C
C             END   CALCULATIONS FOR STEPPED OR MIXED SEALS ONLY
C
C             CALCULATION OF ALPHA FOR STRAIGHT AND MIXED SEALS
C
440   IF (TYPE.EQ.STEP.OR.ALPH(IP1).GT.0.) GO TO 450
      ALPH(IP1)=0.
      IF (I.EQ.NOKNIF) GO TO 450
      XX=(AKP(IP1)-(AKT(I)+AKT(IP1))/2.)/AKH(IP1)
      IF (NOKNIF.GT.1) ALPH(IP1)=3.79*BSQRT(XX)
C
450   CONTINUE
      TANAL(IP1)=BTAN(ALPH(IP1))
      IF (ALPH(IP1).GT.0.0) OOTANA(IP1)=1./TANAL(IP1)
      THETA=AKTH(I)
      BETA=AKBETA(I)/2.
      BETAC=90.-BETA
      TANBET=BTAN(BETA)
      IF (ALENGH.GT.0.0) AREA=CL(I)*ALENGH
      IF (ALENGH.LE.0..AND.NOSETS.GE.0) AREA=CL(I)*PIE*(DIA(I)+CL(I))
      TT(NL)=TR(I)
      TT(NLP1)=TR(I)
      TT(NLP2)=TR(I)
      TTSR(NL)=BSQRT(TR(I))
      TTSR(NLP1)=TTSR(NL)
      TTSR(NLP2)=TTSR(NL)
      A(NL)=AREA
      A(NLP1)=AREA
      A(NLP2)=AREA
      XK(NL)=0.0
      XK(NLP1)=0.0
      XK(NLP2)=1.0
      XKPRT(NL)=0.0
      XKPRT(NLP1)=0.0
      XKPRT(NLP2)=1.0
      XKPRTP(NL)=0.0
      XKPRTP(NLP1)=0.0
      XKPRTP(NLP2)=1.0
      KURVE(NL)=2
      KURVE(NLP1)=1
```

```
      KURVE(NLP2)=0
      IDPORQ(NL)=1
      IDPORQ(NLP1)=1
      IDPORQ(NLP2)=0
      PKR(I)=ARFACT*AKR(I)
      PARM(NL)=ARFACT*AKR(I)/CL(I)
      PARM(NLP1)=AKT(I)/CL(I)
      PARM(NLP2)=0.0
      ADDER(NL)=0.0
      ADDER(NLP1)=0.0
      ADDER(NLP2)=0.0
      IF (THETA.LT.90.) ADDER(NL)=2.73*(1.-BSIN(THETA))
      DELTAC=0.0
      DELTAE=0.0
      IF (I.LE.1.OR.KCCO(I).LT.0) GO TO 460
      IF (ALPH(I).LE.0.0) GO TO 460
      IF (THETA.GT.BETAC) DELTAC=(AKP(I)-0.5*(AKT(I)+AKT(IM1)))/(OOTANA
     1(I)+TANBET)
      IF (THETA.LE.BETAC) DELTAC=(AKP(I)-0.5*(AKT(I)+AKT(IM1)))*TANAL(I)
      IF (DELTAC.GT.AKH(I)) DELTAC=AKH(I)
460   IF (I.GE.NOKNIF.OR.KECO(I).LT.0) GO TO 470
      IF (ALPH(IP1).LE.0.0) GO TO 470
      THETAP=AKTH(IP1)
      BETAP=AKBETA(IP1)/2.
      BETAPC=90.-BETAP
      TANBTP=BTAN(BETAP)
      IF (THETAP.GT.BETAPC) DELTAE=(AKP(IP1)-0.5*(AKT(I)+AKT(IP1)))/
     1(OOTANA(IP1)+TANBTP)
      IF (THETAP.LE.BETAPC) DELTAE=(AKP(IP1)-0.5*(AKT(I)+AKT(IP1)))
     1*TANAL(IP1)
      IF (DELTAE.GT.AKH(IP1)) DELTAE=AKH(IP1)
470   AMULTC=1.0
      AMULTE=1.0
      IF (DELTAC.GT.0.0) AMULTC=1.-CL(I)/(CL(IM1)+DELTAC)
      IF (DELTAE.GT.0.0) AMULTE=1.-CL(I)/(CL(I)+DELTAE)
      IF (AMULTC.LT.0.0) AMULTC=0.000001
      IF (AMULTE.LT.0.0) AMULTE=0.000001
      AMUL(NL)=AMULTC
      FETH=THETA
      IF (THETA.EQ.90.) FETH=THETA+BETA
      IF (FETH.GT.90.) AMUL(NL)=AMUL(NL)*(1.-.014*(FETH-90.))
      AMUL(NLP1)=BSQRT(AMULTC)*AMULTE
      AMUL(NLP2)=AMULTE**2
      DELC(I)=DELTAC
      DELE(I)=DELTAE
C
C          SET UP ROUGHNESS ARRAY VARIABLES
C
      ROUFLD(NL)=0.0
      ROUFLD(NLP1)=0.0
      ROUFLD(NLP2)=0.0
      ROUFED(NL)=0.0
      ROUFED(NLP1)=0.0
      ROUFED(NLP2)=0.0
      ROUFDH(NL)=0.0
      ROUFDH(NLP1)=0.0
      ROUFDH(NLP2)=0.0
      IF (ROUGH(I).LE.30.) GO TO 490
      ROUFDH(NLP1)=2.0*CL(I)
C             FOR STEP    SEAL LENGTH PARM. = KT
C             FOR STAIGHT SEAL LENGTH PARM. = KT FOR LAST KNIFE
      DUM=AKT(I)
      IF (TYPE.EQ.STEP.OR.I.EQ.NOKNIF) GO TO 480
```

```
      IF (TYPE.EQ.TMIX.AND.ASH(IP1).NE.0.0) GO TO 480
C               FOR STAIGHT SEAL LENGTH PARM. = KP FOR ALL BUT LAST KNIFE
      DUM=AKP(IP1)
480   ROUFLD(NLP1)=DUM/ROUFDH(NLP1)
      ROUFED(NLP1)=1.D-6*(ROUGH(I)-30.)/ROUFDH(NLP1)
490   CONTINUE
C
C           END SETTING UP ROUGHNESS ARRAY VARIABLES
C
C           SET UP FOR FRICTION FACTOR READ IN
C
      FFLD(NL)=0.0
      FFLD(NLP1)=0.0
      FFLD(NLP2)=0.0
      IF (A4FLOD(I).LE.0.0) GO TO 500
      FFLD(NLP1)=A4FLOD(I)
      AMUL(NLP1)=0.0
      ADDER(NLP1)=0.0
      PARM(NLP1)=0.0
      KURVE(NLP1)=0
500   CONTINUE
C
C     *****************      END   LOOP OVER EACH KNIFE    *************
C
      NL1=NL+2
      TT(1)=TR(1)
      TTSR(1)=BSQRT(TR(1))
      A(1)=A(2)
C
C        DETERMINE MINIMUM AREA TO REFERENCE PHI IF AOPT.NE.AOPT1(UNITY)
C              'AMIN' IS REFERENCE AREA  USED IN PHI PRINTED AND PUNCHED
C
      AMIN=1.0
      IF (AOPT.EQ.AOPT1) GO TO 550
      IF (ALENGH.LE.0) GO TO 520
      AMIN=A(1)
      DO 510 I=2,NL1
      IF (A(I).LT.AMIN) AMIN=A(I)
510   CONTINUE
      GO TO 550
520   CLAVG=CL(1)
      IF (NOKNIF.LE.1) GO TO 540
      DO 530 I=2,NOKNIF
530   CLAVG=CLAVG+CL(I)
      CLAVG=CLAVG/NOKNIF
540   AMIN=CLAVG*AVGDIA*PIE
      IF (NOSETS.GE.0) AMIN = CLAVG * (AVGDIA + CLAVG) * PIE
C
550   CONTINUE
C
C             MODIFY AREAS FOR XMUL(I) VALUE AND CL INCREASE DUE TO
C                         ROUGHNESS
C
C                 AA'S ARE ACTUAL AREAS AND A'S ARE MODIFIED AREAS USED
C                           IN FLOW CALCULATIONS
C
      DO 560 I=1,NL1
560   AA(I)=A(I)
      IF (NOSETS.GE.0.AND.TYPE.NE.STRGHT) PARVAL=XMUL(2)
      DUM=1.0
      IF (ROUGH(1).GT.30.0) DUM=(CL(1)+ROUGH(1)*0.000001)/CL(1)
      A(1)=A(1)*XMUL(1)*DUM
      J=1
```

```
      DO 570 I=2,NL1,3
      DUM=1.0
      IF (ROUGH(J).GT.30.0) DUM=(CL(J)+ROUGH(J)*0.000001)/CL(J)
      A(I)=A(I)*XMUL(J)*DUM
      A(I+1)=A(I+1)*XMUL(J)*DUM
      A(I+2)=A(I+2)*XMUL(J)*DUM
      J=J+1
570   CONTINUE
C
C        PRINT INPUT INFORMATION
C
      IF (NOKNRC.LE.1.AND.NOSETS.GE.0) GO TO 610
C     WRITE (6,580)
      WRITE (6,590)
580   FORMAT (1H)
590   FORMAT (/,4X,'K N I F E -- T O -- K N I F E   S E A L   D E S I',
     1             ' G N   M O D E L')
      WRITE (6,600) TITLE,TITLEC
600   FORMAT (/,4X,20A4,/,4X,20A4)
610   I=1
      IF (TYPE.EQ.STEP) I=2
      IF (TYPE.EQ.TMIX) I=3
      WRITE (6,620) G,XMW,NOKNIF,TYP(I),DIRECT,ALENGH,AVGDIA,PT(1)
620   FORMAT (1H /1H ,'SPECIFIC HEAT RATIO   (GAMMA)  =',F10.4/1H ,'MOLE
     1CULAR WEIGHT               =',F10.4/1H ,'NUMBER OF KNIVES
     2      =',I10/1H ,'SEAL TYPE                        =',2X,A8/1H ,'FLO
     3W DIRECTION                =',6X,A4/1H ,'SEAL LENGTH (2-D SEAL)
     4       =',F10.4,' (INCHES)'/1H ,'AVG. KNIFE DIAMETER (3-D SEAL) ='
     5,F10.4,' (INCHES)'/1H ,'INLET TOTAL PRESSURE           =',F10.4,'
     6(PSIA)')
      WRITE (6,630)
630   FORMAT (/,4X,'K N I F E   G E O M E T R Y   D A T A', /,' KNIFE
     1CL     KR      KT     KP     KH     SH     DTC  ','THETA  BETA   D
     2IA   ROUGH  TEMP  KCCO KECO 4FL/D ',' DEL C  DEL E  AREA  ALPHA')
      WRITE (6,640)
640   FORMAT (1H ,'  NO.  (IN)   (IN)    (IN)   (IN)   (IN)   (IN)   (IN
     1)  ','(DEG) (DEG)  (IN)   (RMS) (DEGR)                  ','  (IN)
     2 (IN)  MULT  (DEG)')
      WRITE (6,650)
650   FORMAT (' ----- ------ ------- ------ ------ ------ ------ ------
     1','----- ----- ------  ----- ------ ---- ---- ----- ','------ ----
     2-- -----  -----'/)


      DO 680 I=1,NOKNIF
      IF ( I .EQ. 1 )
     +WRITE(6,660) I,CL(I),PKR(I),AKT(I),AKP(I),AKH(I),ASH(I),ADTC(I)
     1,AKTH(I),AKBETA(I),DIA(I),ROUGH(I),TR(I),KCCO(I),KECO(I),A4FLOD(I)
     2,DELC(I),DELE(I),XMUL(I)
      IF ( I .GT. 1 )
     +WRITE(6,660) I,CL(I),PKR(I),AKT(I),AKP(I),AKH(I),ASH(I),ADTC(I)
     1,AKTH(I),AKBETA(I),DIA(I),ROUGH(I),TR(I),KCCO(I),KECO(I),A4FLOD(I)
     2,DELC(I),DELE(I),XMUL(I),ALPH(I)
  660 FORMAT(I5,F8.4,F8.5,5F7.4,2F6.1,F7.4,F7.2,F7.1,I4,I5,F8.2,2F7.4,
     1          2F6.3)
680   CONTINUE


      IF (NOSETS.LT.0.OR.NOKNRC.GT.1) GO TO 690
      CALL INCHEK (CL,AKT,AKP,AKH,ASH,ADTC,AKTH,ROUGH,TYPE,NOKNRC,IERR
     1,NOKNIF,DIRECT)
C
C        CALL SUBROUTINE CHOKEC  TO FIND CHOKE POINT OR USE W'S READ IN
```

```
C
690   NQ=IPRINT
      MG=ICVTYP
      NV=NOW
      IF (NV.GT.0) NQ=1
      CALL CHOKEC (PRATIO,WTRUE,.TRUE.,.FALSE.)
C
C        CALL SUBROUTINE FCURVE TO CAL. CURVE IF ICURVE > 0
C                  AND ERROR NOT ENCOUNTERED IN CHOKEC
C
      IF (ICURVE.LE.0.OR.KODE.EQ.3) GO TO 720
      IF (TYPE.EQ.STEP) GO TO 710
      PARVAL=0.0
      IF (NOKNIF.LE.1) GO TO 710
      PARVAL=101.
      DO 700 I=1,NOKNIF
      IF (KCCO(I).GE.0) PARVAL=ALPH(I)
      IF (KECO(I).GE.0) PARVAL=ALPH(I)
700   CONTINUE
710   CALL FCURVE (NOKNIF,PRATIO,.TRUE.)
720   IF (NOSETS.LE.0) GO TO 50
      IF (ISETS.GE.NOSETS) GO TO 10
      GO TO 70
730   IF (TYPE.NE.TMIX) GO TO 750
      WRITE (6,740)
740   FORMAT (' MIXED TYPE SEAL CANNOT BE OPTIMIZED WITH THIS PROGRAM')
      GO TO 760
750   CALL OPTSUB
      GO TO 720
760   STOP
      END
C ---------------------------------------------------------------------
C
C     Subroutine FILES to generate OUTPUT file names
C
C ---------------------------------------------------------------------
      SUBROUTINE FILES(CH30,LEN,FILN)
      CHARACTER  CH1*1
      CHARACTER  CH30*80, FILN(5)*80, CH4(5)*4
      DATA CH4 /'.INP','.OUT','.FIL','.   ','.   '/
C
      LEN = LEN+1
      CH30(LEN:LEN) = ' '
      DO 10 I = 1, LEN
      CH1 = CH30(I:I)
      IF (CH1 .EQ. ' ' .OR. CH1 .EQ. '.') GO TO 20
   10 CONTINUE
   20 CONTINUE
      J = I
      LST = I + 3
C
C ********************************************************************
C THE DO LOOP WAS CHANGED TO LEAVE INPUT FILE NAME IN FILN(1)
C UNCHANGED
C ********************************************************************
      DO 30 I = 2, 5
      FILN(I)         = ' '
      FILN(I)(1:J)    = CH30(1:J)
      FILN(I)(J:LST) = CH4(I)
   30 CONTINUE
      RETURN
      END
C ---------------------------------------------------------------------
```

```
C
C     Block data statement to store data
C
C -------------------------------------------------------------
      BLOCK DATA
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER     TMIX*5, TITLE*4, TITLEC*4, AOPT*5, XTYPE*5, AOPTD*5
      COMMON /C12R/ WVAL(30),TT(100),AA(100)
      COMMON /C13R/ PARVAL,AMIN,ROGCN
      COMMON /C13RA/TITLE(20),TITLEC(20),AOPTD
      COMMON /C13I/ MG,NOSETS,ISETS
      COMMON /C14R/ G,GP1,GP1GM1,GP1OG,GP1O2,GM1O2,XK(100)
      COMMON /C15R/ C1,GP1GM2,CMAX,GM1O2D,GOGM1
      COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
     1,XKPRT(100),XKPRTP(100)
      COMMON /C124I/ KURVE(100),IDPORQ(100)
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /C1234I/ NL1,KODE
C     COMMON /OPTDAT/ ALENGH,DMIN,DMAX,ALPHA,AOPT,PRATIO,XMW,ARFACT,NOW
C    1,ICURVE,ICVTYP,IPRINT
      COMMON /C78R/ XCL,XKP,XKH,XKT,XSH,XDTC,XDIRCT,XKR,XKTH
     1,XVGDIA,XELDIA,XLENGH,XKBETA
      COMMON /C78RA/ XTYPE
      COMMON /OPTDAT/ ALENGH,DMIN,DMAX,ALPHA,PRATIO,XMW,ARFACT,NOW
     1,ICURVE,ICVTYP,IPRINT
      COMMON /OPTX/ AOPT
      COMMON /VECTR/ CL(30),AKR(30),AKT(30),AKP(30),AKH(30),ASH(30),ADTC
     1(30),AKTH(30),AKBETA(30),DIA(30),ROUGH(30),TR(30),A4FLOD(30),DELC
     2(30),DELE(30),KCCO(30),KECO(30),PKR(30)
      COMMON /CROUGH/ ROUFLD(100),ROUFED(100),ROUFDH(100)
      COMMON /CMIX/ TMIX
C
C
      DATA PARVAL /0.0/
      DATA A/100*0.0/, FFLD/100*0.0/, PARM/100*0.0/
      DATA AMUL/100*0.0/, ADDER/100*0.0/, XKPRT/100*0.0/
      DATA XKPRTP/100*0.0/

      DATA KURVE/100*0.0/, IDPORQ /100*0.0/

      DATA PT/100*0.0/, TTSR /100*0.0/

      DATA WVAL/30*0.0/, TT/100*0.0/, XK/100*0.0/
      DATA TR  /30*0.0/, DELC/30*0.0/, DELE/30*0.0/, PKR/30*0.0/

      DATA ROUFLD/100*0.0/, ROUFED/100*0.0/, ROUFDH/100*0.0/

      DATA  TMIX/'MIXED'/
      END
C -------------------------------------------------------------
C
C                          SUBROUTINE     CHOKEC
C
C            CALCULATION OF CHOKE POINT USING INTERVAL HALVING
C
C            OR USE W'S READ IN TO DETERMINE FLOW CONDITIONS
C
C -------------------------------------------------------------
      SUBROUTINE CHOKEC (PRATIO,WTRUE,PRNT,OPT)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL PRNT,OPT
C
      COMMON /C12R/ WVAL(30),TT(100),AA(100)
```

```
      COMMON /C12I/ NV,NQ
      COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
     1,XKPRT(100),XKPRTP(100)
      COMMON /C124I/ KURVE(100),IDPORQ(100)
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /C1234I/ NL1,KODE
      COMMON /C234R/ V,W,FF,PS(100),XMN(100)
C
      CHARACTER*8   TDPORQ(3), TYPLOS(3)
      DATA TDPORQ/'        ','  PT-PS ','    Q   '/
      DATA TYPLOS/' EXPND',' LHOLE',' CONTR'/
      BABS(ADUM)=ABS(ADUM)
C
      IF (.NOT.OPT) ICMAX=30
      IF (.NOT.OPT) DELMAX=.3E-6
      IF (OPT) ICMAX=45
      IF (OPT) DELMAX=.1E-11
      NV1=0
      W=0.
10    NV1=NV1+1
      IF (NV.GT.0.AND.NV1.GT.NV) GO TO 220
      IF (NV.GT.0) W=WVAL(NV1)
      IF (W.GT.0.0) GO TO 30
      AMIN=A(1)/TTSR(1)
      DO 20 I=2,NL1
      DUM=A(I)/TTSR(I)
      IF (DUM.LT.AMIN) AMIN=DUM
20    CONTINUE
      W=0.532*AMIN*PT(1)
C
C     ********** START OF ITERATION ************************************
C
30    IPDA=0
      ICOUNT=0
      WF=0.0
      WB=0.0
40    ICOUNT=ICOUNT+1
C
C     ********** CALL SUBROUTINE FLOWCL       *********
C
      CALL FLOWCL
      IF (PRATIO.LE.0.) GO TO 70
C                                  ITERATION FOR DESIRED PRESSURE RATIO
      IF (KODE.EQ.3) GO TO 80
      IF (KODE.EQ.2) GO TO 50
      RAT=PT(1)/PT(NL1)
      IF (ICOUNT.GT.ICMAX) GO TO 180
      IF (RAT.LT.PRATIO) GO TO 60
C                                          FLOW TOO HIGH
50    WF=W
      W=.5*(WF+WB)
      GO TO 170
C                                          FLOW TOO LOW
60    IF (NQ.GT.0) GO TO 100
      WB=W
      WQP=W
      IF (WF.LE.0.) W=2.*WB
      IF (WF.GT.0.) W=.5*(WF+WB)
      GO TO 170
C                                       END PRESSURE RATIO ITERATION
70    GO TO (100,90,80), KODE
C
C     ******* BRANCH FOR CHOKED FLOW ENCOUNTERED ******
```

```
C
80    WRITE (6,290)
      GO TO 220
90    CONTINUE
      IF (NQ.GT.0) WRITE (6,230) W
      IF (NQ.GT.0) WRITE (6,240)
      IF (IPDA.GT.0) GO TO 220
      IF (NV.GT.0) GO TO 10
      IF (ICOUNT.GT.ICMAX) GO TO 180
      WF=W
      IF (WB.LE.0.0) W=0.5*WF
      IF (WB.GT.0.0) W=0.5*(WF+WB)
      GO TO 170
C
C     ******* BRANCH FOR NON-CHOKED FLOW ******
C
100   IF (NQ.LE.0) GO TO 160
      IF (.NOT.PRNT) GO TO 110
      IF (NV.LE.0.AND.PRATIO.LE.0.) WRITE (6,250) W,ICOUNT
      IF (NV.GT.0.OR.PRATIO.GT.0.) WRITE (6,260) W,ICOUNT
110   WTRUE=W
      IF (.NOT.PRNT) GO TO 150
      WRITE (6,270)
      DO 140 I=1,NL1
      K=KURVE(I)+1
      KN=(I+1)/3
      FFS=W*TTSR(I)/A(I)
      FF=FFS/PT(I)
      IF (I.LT.NL1) FFS=FFS/PS(I)
      J=2
      IF (IDPORQ(I).GT.0) J=3
      IF (XKPRT(I).EQ.0.0) J=1
      IF (I.LE.1) WRITE (6,120) I,AA(I),TT(I),FF,FFS,PT(I),PS(I),XMN(I)
     1,A(I)
120   FORMAT (I5,5X,F9.3,F8.2,F8.4,F8.4,17X,F10.3,F8.3,F6.3,35X,F9.3)
      IF (I.GT.1.AND.I.LT.NL1) WRITE (6,280) I,KN,AA(I),TT(I),FF,FFS
     1,FFLD(I),XKPRT(I),TYPLOS(K),PT(I),PS(I),XMN(I),TDPORQ(J),PARM(I)
     2,AMUL(I),ADDER(I),XKPRTP(I),A(I)
      IF (I.EQ.NL1) WRITE (6,130) I,KN,AA(I),TT(I),FF,FFLD(I),XKPRT(I)
     1,TYPLOS(K),PT(I),TDPORQ(J),PARM(I),AMUL(I),ADDER(I),XKPRTP(I),A(I)
130   FORMAT (2I5,F9.3,F8.2,F8.4,8X,F7.3,F6.3,A6,F8.3,14X,A7,F7.3,3F7.3,
     1F9.3)
140   CONTINUE
150   FF=W*TTSR(1)/PT(1)
      V=PT(1)/PT(NL1)
160   IF (IPDA.GT.0) GO TO 220
      IF (NV.GT.0) GO TO 10
      WQP=W
      IF (ICOUNT.GT.ICMAX) GO TO 180
      WB=W
      IF (WF.LE.0.0) W=2.0*WB
      IF (WF.GT.0.0) W=0.5*(WF+WB)
170   IF (WF.LE.0.0) GO TO 40
      IF ((BABS(WF-WB)/WF).GT.DELMAX) GO TO 40
C
C     **********   END OF ITERATION   **********************************
C
180   CONTINUE
      IF (PRATIO.LE.0.) GO TO 210
      IF (PRNT) WRITE (6,190) W,RAT,PRATIO
190   FORMAT (' ',T20,'FLOW',G14.6,5X,'PRESSURE RATIO',G14.6,5X,'DESIRED
     1 PR',G14.6)
      IF (KODE.EQ.2.OR.(PRATIO-RAT).GT..001) WRITE (6,200)
```

```
200   FORMAT (T20,'DESIRED PRESSURE RATIO COULD NOT BE OBTAINED WITHOUT
     1EXCEEDING CHOKED FLOW')
210   IF (NQ.GT.0) GO TO 220
C
C     **   INITIATE A LAST PASS IF PRINT(NQ) WAS NOT TURNED ON   *******
C
      IPDA=1
      W=WQP
      NQ=1
      GO TO 40
220   RETURN
C
C
230   FORMAT (//5X,2HW=,F10.5,'LBS/SEC   THIS AMOUNT OF FLOW GIVES A VAL
     1UE BEYOND CHOKE'/)
240   FORMAT ('  X X X X X X X X X X X X X X X X X X X X X X X X X X X X
     1 X X X X X X X X X X .')
250   FORMAT (/,4X,   'F L O W   R E S U L T S   A T   C H O K E   P O I
     1 N T',5X,'(W =',F9.5,' LB/SEC   ITERATIONS =',I3,')')
260   FORMAT (/,4X,   'F L O W   R E S U L T S   A T   S P E C I F I E D
     1   F L O W    R A T E',5X,'(W =',F9.5,' LB/SEC   ITERATIONS =',I3,
     2')')
270   FORMAT (' ',' STN KNIFE   AREA    TEMP   WRT/PTA WRT/PSA ','4FL/D
     1KFACT  LOSS    PT      PS     MN  KFACT','    PARM   AMUL  ADDER
     2XKUNC  MOD AREA '/'  NO.  NO.  (IN**2) (DEG R)',T59,'TYPE',T65,'(P
     3SIA)  (PSIA)',T86,'METHOD',T122,'(IN**2)'/'  --- ----- ------- ---
     4----  ------- ------- ','----- ----- -----  ------  ------ ----- -
     5-----','  -----  -----  -----  -----  --------'/)
280   FORMAT (2I5,F9.3,F8.2,F8.4,F8.4,F7.3,F6.3,A6,F8.3,F8.3,F6.3,A7,F7.
     13,3F7.3,F9.3)
290   FORMAT (10X,'***** ERROR  -  ADJACENT AREAS NOT EQUAL FOR A FRICTI
     1ON TYPE LOSS *****')
      END
C -------------------------------------------------------------------
C
C                          SUBROUTINE     FCURVE
C
C                 CALCULATION OF FLOW CURVE GIVEN CHOKE POINT
C
C -------------------------------------------------------------------
      SUBROUTINE FCURVE (NOKNIF,PRATIO,PRNT)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL PRNT
      CHARACTER   AOPTD*5, XTYPE*5, AOPT1*5, TITLE*4, TITLEC*4

      DIMENSION FLOF(15), RET(15), FSAV(15), ROGTFL(15)
C
      COMMON /C13R/  PARVAL,AMIN,ROGCN
      COMMON /C13RA/ TITLE(20),TITLEC(20),AOPTD
      COMMON /C13I/ MG,NOSETS,ISETS
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /C1234I/ NL1,KODE
      COMMON /C234R/ V,W,FF,PS(100),XMN(100)
      COMMON /ERRCAL/ IFLAG(100),X(100),Y(100),NP
      COMMON /C78R/ XCL,XKP,XKH,XKT,XSH,XDTC,XDIRCT,XKR,XKTH
     1,XVGDIA,XELDIA,XLENGH,XKBETA
      COMMON /C78RA/ XTYPE
      DATA AOPT1/'UNITY'/
      BSQRT(SDUM)=SQRT(SDUM)
C
      FG=FF/AMIN
      AY=0.13
      QP=0.11
```

```
      AQ=W
      NPTS=14
C
C     ********** START LOOP OVER FLOW RATE ******************************
C
      IF (PRATIO.GT.0.) GO TO 50
      DO 40 J=2,13
      IF (J.GT.2) QP=QP+(AY-((J-3)/100.0))
      W=AQ*QP
C
C     ********** CALL SUBROUTINE FLOWCL       *********
C
      CALL FLOWCL
      GO TO (30,10,20), KODE
C
10    WRITE (6,260)
      WRITE (6,270) W
      NPTS=J
      RET(J)=V
      FLOF(J)=W*TTSR(1)/PT(1)/AMIN
      GO TO 50
20    WRITE (6,280)
      GO TO 110
30    RET(J)=PT(1)/PT(NL1)
      FLOF(J)=W*TTSR(1)/PT(1)/AMIN
      IF (J.EQ.3.AND.RET(2).LT.1.00001) RET(2)=1.+(RET(3)-1.)*((FLOF(2)
     1/FLOF(3))**2)
40    CONTINUE
C
C     ********** END   LOOP OVER FLOW RATE ******************************
C
C
C     ******* SET UP AND PUNCH FLOW CURVE ***********
C
C
C     ICVTYP (MG)    -  PUNCH CURVE TYP. = 0 NO PUNCHED CURVE
C                                        = 1 PHI    VERSUS PR (DU1  FMT.)
C                                        = 2 PHI    VERSUS PR (PLOT FMT.)
C                                        > 2 PHI    VERSUS PR (BOTH FMT.)
C                       (NOTE: PHI**2 VERSUS 1.-PR**2 PRINTED REGARDLESS
C                              OF ICVTYPE VALUE)
C
C
50    RET(1)=1.0
      RET(14)=V
      RET(15)=5.0*V
      FLOF(1)=0.0
      FLOF(14)=FG
      FLOF(15)=FG
      J1=1
      J2=2
      IF (PRATIO.GT.0.) J1=14
      IF (PRATIO.GT.0.) J2=14
C
C         * * * *  END   LOOP OVER EACH PRESSURE RATIO  * * * *
C
      FLOF(15)=FLOF(14)
      IF (MG.LE.0) GO TO 80
      IF (MG.EQ.2) GO TO 60
C
C        PHI VERSUS PR PUNCHED WITH DU1 PUNCHED OUTPUT FORMAT
C
      IF (PRATIO.GT.0.) GO TO 80
```

```
      J=1
      WRITE (7,120) (TITLE(I),I=1,16),AMIN,J
      WRITE (7,130) RET
      WRITE (7,130) FLOF
      IF (MG.EQ.1) GO TO 80
C
C        PHI VERSUS PR PUNCHED WITH PLOT INPUT FORMAT
C
60    IF (NOSETS.GT.0.AND.ISETS.GT.1) GO TO 70
      IF (NOSETS.LT.0) GO TO 70
      J=NOSETS
      IF (J.LE.0) J=1
      WRITE (7,140) J
      WRITE (7,150) TITLE,(TITLEC(I),I=1,16),AMIN
70    CONTINUE
      IF (PARVAL.LE.100.) WRITE (7,160) PARVAL,NPTS,(RET(I),FLOF(I),I=1
     1,3)
      IF (PARVAL.GT.100.) WRITE (7,170) NPTS,(RET(I),FLOF(I),I=1,3)
      WRITE (7,180) (RET(I),FLOF(I),I=4,NPTS)
      WRITE (7,190)
C
80    DO 90 I=J1,15
      FLOF(I)=FLOF(I)**2
      ROGTFL(I)=ROGCN*FLOF(I)
90    RET(I)=1.-1./RET(I)**2
C
C        PHI**2 VERSUS 1. - 1./PR**2   * * *  P R I N T E D  * * *
C
      IF (.NOT.PRNT) GO TO 100
      IF (AOPTD.EQ.AOPT1) WRITE (6,200)
      IF (AOPTD.NE.AOPT1) WRITE (6,210) AMIN
      WRITE (6,220) (I,I=J1,NPTS)
      WRITE (6,230) (RET(I),I=J1,NPTS)
      WRITE (6,240) (FLOF(I),I=J1,NPTS)
      WRITE (6,250) (ROGTFL(I),I=J1,NPTS)
C
C              END  * * *  P R I N T E D   O U T P U T  * * *
C
100   IF (PRATIO.LE.0.) GO TO 110
      W=BSQRT(FLOF(14))*PT(1)/TTSR(1)
C
110   RETURN
C
120   FORMAT (16A4,'CURVE A=',F6.3,I2)
130   FORMAT (8F10.7)
140   FORMAT (1H ,I17,' LINE(S)',2X,'XX/XX/XX  XXXX')
150   FORMAT (20A4/16A4,'CURVE A=',F6.3)
160   FORMAT (F5.2,1X,I4,10X,2X,F6.3,F7.4,7X,F6.3,F7.4,7X,F6.3,F7.4)
170   FORMAT ('1.E+9',1X,I4,10X,2X,F6.3,F7.4,7X,F6.3,F7.4,7X,F6.3,F7.4)
180   FORMAT (2X,F6.3,F7.4,7X,F6.3,F7.4,7X,F6.3,F7.4,7X,F6.3,F7.4)
190   FORMAT (1H )
200   FORMAT (/,4X,   'F L O W   C U R V E',5X,'WHERE PR = PT UP / PT DO
     1WN  AND PHI = W * SQRT(TIN) / PT UP')
210   FORMAT (/,4X,   'F L O W   C U R V E',5X,'WHERE PR = PT UP / PT DO
     1WN  AND PHI = W * SQRT(TIN) / (PT UP * AREF)',2X,'(WHERE AREF=',F7
     2.3,')')
220   FORMAT (1H ,'I          ',15I8)
230   FORMAT (1H ,'1.-1./PR**2  ',15F8.5)
240   FORMAT (1H ,'PHI**2       ',15F8.5)
250   FORMAT (1H ,'R/G * PHI**2 ',15F8.5)
260   FORMAT (1H ,9X,'***** ERROR  -  CHOKE FLOW IN FCURVE ROUTINE.','
     1LAST POINT HAS CHOKE ROUTINE PR VALUE *****')
270   FORMAT (10X,'W = ',F10.7)
```

```
280   FORMAT (1H ,'***** ERROR  -  ADJACENT AREAS NOT EQUAL FOR A FRICTI
     1ON TYPE LOSS *****')
      END
C -----------------------------------------------------------------
C
C                          SUBROUTINE     FLOWCL
C
C     CALCULATION OF THE FLOW CHARACTERISTICS OF RESTRICTIONS IN
C     SERIES.
C        GIVEN A VALUE OF FLOW RATE:  W
C        KFACT TYPE LOSS CALCULATED ON A PT-PS OR Q BASIS
C
C -----------------------------------------------------------------
      SUBROUTINE FLOWCL
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /C14R/ G,GP1,GP1GM1,GP1OG,GP1O2,GM1O2,XK(100)
      COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
     1,XKPRT(100),XKPRTP(100)
      COMMON /C124I/ KURVE(100),IDPORQ(100)
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /C1234I/ NL1,KODE
      COMMON /C234R/ V,W,FF,PS(100),XMN(100)
      BSQRT(SDUM)=SQRT(SDUM)
      BABS(ADUM)=ABS(ADUM)
      BLOG(FDUM)=DLOG(FDUM)
C
      KODE=2
C
C     ********** START LOOP OVER  EACH LOSS   *********
C
      DO 110 I=1,NL1
      II=I
      IP1=I+1
      IF (I.GT.1.OR.FFLD(IP1).LE.0.) GO TO 10
C
C     *** INITIAL MACH NUMBER CALCULATION  ***
C
      CALL PSCAL (W,A(I),TTSR(I),PT(I),XMN(I),PS(I),NOIT)
      IF (XMN(I).GE.1.) GO TO 120
10    IF (I.GE.NL1) GO TO 110
      IF (FFLD(IP1).GT.0.) GO TO 20
C
C     *** KFACT TYPE LOSS  ***
C
      XKD=XK(IP1)
      KURV=KURVE(IP1)
      CALL PSCAL (W,A(IP1),TTSR(I),PT(I),XMF,PSF,NOIT)
      XMN(I)=XMF
      PS(I)=PSF
      IF (XMF.GE.1.) GO TO 120
      PHI2=W*TTSR(I)/(PT(I)*A(IP1))
      CALL AKCAL (PHI2,PARM(IP1),XKD,KURV,IP1,W,TTSR(I))
      IF (IDPORQ(IP1).LE.0) PT(IP1)=PT(I)-XKD*(PT(I)-PSF)
      IF (IDPORQ(IP1).GT.0) PT(IP1)=PT(I)-XKD*(G/2.*PSF*XMF**2)
      IF (PT(IP1)) 120,120,110
C
C     *** 4FL/D TYPE LOSS  ***
C
20    PTR=BSQRT((2.*(1.+GM1O2*XMN(I)**2)/(GP1))**GP1GM1)/XMN(I)
      IF (A(IP1).EQ.A(I)) GO TO 30
      KODE=3
      WRITE (6,130) A(I),I,A(IP1),IP1
      GO TO 120
```

```
30     CONTINUE
       PSR=BSQRT(GP1O2/(1.+GM1O2*XMN(I)**2))/XMN(I)
       FFLDM=(1.-XMN(I)**2)/G/XMN(I)**2+GP1OG/2.*BLOG((GP1)*XMN(I)**2/2./
      1(1.+GM1O2*XMN(I)**2))
       FFLDM=FFLDM-FFLD(IP1)
       IF (FFLDM) 120,120,40
40     AM=XMN(I)
       LOOP=0
50     FFL2=(1.-AM**2)/G/AM**2+GP1OG/2.*BLOG((GP1)*AM**2/2./(1.+GM1O2*AM*
      1*2))
       IF (BABS(FFLDM-FFL2)/FFLDM-.0001) 100,100,60
60     IF (LOOP.GT.0) GO TO 70
       FFL1=FFL2
       AM1=AM
       AM=(XMN(I)+1.)/2.
       LOOP=LOOP+1
       GO TO 50
70     AM2=AM
       AM=AM2-(AM2-AM1)/(FFL2-FFL1)*(FFL2-FFLDM)
       IF (AM) 80,80,90
80     AM=AM2/2.
90     AM1=AM2
       FFL1=FFL2
       GO TO 50
100    XMN(IP1)=AM
       PTW=BSQRT((2.*(1.+GM1O2*XMN(IP1)**2)/(GP1))**GP1GM1)/XMN(IP1)
       PSW=BSQRT(GP1O2/(1.+GM1O2*XMN(IP1)**2))/XMN(IP1)
       PT(IP1)=PTW/PTR*PT(I)
       PS(IP1)=PSW/PSR*PS(I)
110    CONTINUE
       KODE=1
120    RETURN
130    FORMAT (2X,5('**ERROR**'),/2X,'A FRICTION LOSS MAY NOR HAVE',' AN
      1AREA CHANGE ACROSS IT',/2X,'CHECK YOUR INPUT',10X,'A=',F10.4,' AT
      2STATION',I5,'A=',F10.4,' AT STATION',I5)
       END
C ---------------------------------------------------------------------
C
C                          SUBROUTINE     AKCAL
C
C                SUBROUTINE TO CALCULATE LOSS COEFFICIENTS FOR
C                             KTK SEAL MODEL
C
C ---------------------------------------------------------------------
       SUBROUTINE AKCAL (PHI,PAR,XK,KURVE,IP1,W,TTSR)
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
      1,XKPRT(100),XKPRTP(100)
       COMMON /CROUGH/ ROUFLD(100),ROUFED(100),ROUFDH(100)
C
       DIMENSION AK(10,6), PHIT(10), PARMM(6), C(5)
       BEXP(EDUM)=EXP(EDUM)
       BLOG(FDUM)=DLOG(FDUM)
C
C              NDX    =    1ST DIMENSION OF AK
C
       DATA NDX/10/
C
C              CONSTANTS FOR K CONTRACTION  -  USED FOR KURVE = 2
C
       DATA C/0.700,-0.333,-4.600,0.250,1.118/
C
C              K LONG HOLE DATA DERIVED FROM KEARTON AND KEH
```

```
C                  USED FOR KURVE = 1
C
      DATA NOPHI/10/,NOPAR/6/
      DATA PHIT/0.00,0.25,0.30,0.40,0.44,0.449,0.46,0.492,0.4944,1.000/
C
      DATA PARMM/0.77,1.58,3.28,4.00,6.00,12.0/
      DATA AK/1.20,1.03,0.95,0.76,0.68,0.651,0.576,0.291,0.27,0.27,0.90,
     10.83,0.79,0.67,0.63,0.651,0.576,0.291,0.27,0.27,0.27,0.27,0.27,0.2
     27,0.27,0.27,0.27,0.27,0.27,0.27,0.18,0.18,0.18,0.18,0.18,0.18,0.18
     3,0.18,0.18,0.18,0.06,0.06,0.06,0.06,0.06,0.06,0.06,0.06,0.06,0.06,
     40.12,0.12,0.12,0.12,0.12,0.12,0.12,0.12,0.12,0.12/
C
C             CONSTANTS FOR ROUGHNESS FOURF MULTIPLIER
C
      DATA ROUFC1 /6.02/, ROUFC2 /-138.41/
C
C
C             K LONG HOLE CALCULATION (TABLE LOOKUP)
C                  USED FOR KURVE = 1
C
      IF (KURVE.NE.1) GO TO 20
      CALL MTABLU (PHIT,AK,PARMM,PHI,XK,PAR,NOPHI,NOPAR,NDX,1,0,1)
C
C                ACCOUNTING FOR ROUGHNESS IN KLH IF ROUGH > 30
C
      IF (ROUFED(IP1).LE.0.0) GO TO 10
      TEMP=TTSR**2
      WA=W/A(IP1)
      XMU=0.002629*TEMP**1.5/(TEMP+198.6)
      XNRES=43200.*WA*ROUFDH(IP1)/XMU
      FOURF=1./(0.825*BLOG(10./XNRES+0.2*ROUFED(IP1)))**2
      FOURFO  =  1./(0.825 * BLOG (10. / XNRES ))**2
      DUM     =  ROUFC1 + ROUFC2 * ROUFED(IP1)
      IF (DUM.LT.0.0) DUM = 0.0
      ADDER(IP1) = (FOURF - FOURFO) * ROUFLD(IP1) * DUM
10    CONTINUE
      IF (XK.LT.0.0) XK=0.0
      XKPRTP(IP1)=XK
      IF (AMUL(IP1).GT.0.0) XK=XK*AMUL(IP1)
      XK=XK+ADDER(IP1)
      GO TO 40
C
C             CALCULATION OF K CONTRACTION  -  USED FOR KURVE = 2
C
20    IF (KURVE.NE.2) GO TO 30
      XK=C(1)
      IF (PAR.LE.0.0) GO TO 30
      XK=C(1)*(1.-BEXP(C(2)+C(3)*PHI**2/PAR**C(4)+C(5)*PAR))
30    CONTINUE
      IF (XK.LT.0.0) XK=0.0
      XKPRTP(IP1)=XK
      XK=XK+ADDER(IP1)
      IF (AMUL(IP1).GT.0.0) XK=XK*AMUL(IP1)
C
40    IF (XK.LT.0.0) XK=0.0
      XKPRT(IP1)=XK
C
      RETURN
      END
C ---------------------------------------------------------------------
C
C                          SUBROUTINE     PSCAL
C
```

```
C                SUBROUTINE TO CALCULATE MACH NUMBER AND PS
C
C ---------------------------------------------------------------------
      SUBROUTINE PSCAL (W,A,SRT,PT,XMN,PS,IT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /C15R/ C1,GP1GM2,CMAX,GM1O2D,GOGM1
C
      DATA TOL/0.1E-11/,BUMP/0.01/,DXMX/0.1/,ITMX/100/
      BSQRT(SDUM)=SQRT(SDUM)
      BABS(ADUM)=ABS(ADUM)
C
      Q=W*SRT/(PT*A)
      IF (Q.GE.CMAX) GO TO 50
      XMN=1.-BSQRT(1.-Q/CMAX)
      DO 30 IT=1,ITMX
      QC=C1*XMN/(1.+GM1O2D*XMN**2)**GP1GM2
      IF (QC.LE.0.0) GO TO 40
      XZX1=Q/QC
      IF (BABS(XZX1-1.0).LE.TOL) GO TO 70
      IF (IT.GT.1) GO TO 10
      DXMN=XMN*BUMP
      GO TO 20
10    IF ((BABS(XMNO-XMN)/XMN).LE.TOL) GO TO 70
      IF (XZXO.EQ.XZX1) GO TO 60
      DXMN=(XMNO-XMN)/(XZXO-XZX1)*(XZX1-1.0)
20    IF (BABS(DXMN/XMN).GT.DXMX) DXMN=BABS(DXMX*XMN/DXMN)*DXMN
      XMNO=XMN
      XZXO=XZX1
      XMN=XMN-DXMN
30    CONTINUE
      IT=ITMX+1
      GO TO 70
40    XMN=0.0
      PS=PT
      GO TO 80
50    XMN=1.0
      GO TO 70
60    IT=IT+2000
70    PS=PT/(1.+GM1O2D*XMN**2)**GOGM1
80    RETURN
      END
C ---------------------------------------------------------------------
C
C                          SUBROUTINE     TABLU
C
C           SUBROUTINE TO DO A SINGLE TABLE LOOK UP FOR Y GIVEN X
C                WITH EXTRAPOLATION PERMISSIBLE FOR IEX.LE.0
C                Y SQUARED       CONSIDERED FOR ISQ.EQ.1
C                LOG X           CONSIDERED FOR ISQ.EQ.2
C                LOG X AND LOG Y CONSIDERED FOR ISQ.GE.3
C
C ---------------------------------------------------------------------
      SUBROUTINE TABLU (XT,YT,X,Y,N,IEX,ISQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XT(1), YT(1)
      BSQRT(SDUM)=SQRT(SDUM)
      BLOG(FDUM)=DLOG(FDUM)
      BEXP(EDUM)=EXP(EDUM)
C
      IDIR=1
      IMIN=1
      IMAX=N
      IF (XT(N).GE.XT(1)) GO TO 10
```

```
      IDIR=-1
      IMIN=N
      IMAX=1
10    A=X
      IF (IEX.GT.0.AND.A.LT.XT(IMIN)) A=XT(IMIN)
      IF (IEX.GT.0.AND.A.GT.XT(IMAX)) A=XT(IMAX)
      DO 20 I=2,N
      DIF=XT(I)-A
      IF (IDIR.GT.0.AND.DIF.GE.0.0) GO TO 30
      IF (IDIR.LT.0.AND.DIF.LE.0.0) GO TO 30
20    CONTINUE
      I=N
30    I1=I-1
      Y2=YT(I)
      Y1=YT(I1)
      X2=XT(I)
      X1=XT(I1)
      IF (ISQ.EQ.1) Y2=Y2**2
      IF (ISQ.EQ.1) Y1=Y1**2
      IF (ISQ.GE.2) X2=BLOG(X2)
      IF (ISQ.GE.2) X1=BLOG(X1)
      IF (ISQ.GE.3) Y2=BLOG(Y2)
      IF (ISQ.GE.3) Y1=BLOG(Y1)
      IF (ISQ.GE.2) DIF=X2-BLOG(A)
      Y=Y2-DIF*(Y2-Y1)/(X2-X1)
      IF (ISQ.EQ.1) Y=BSQRT(Y)
      IF (ISQ.GE.3) Y=BEXP(Y)
      RETURN
      END
C -----------------------------------------------------------------
C
C                          SUBROUTINE     MTABLU
C
C      SUBROUTINE TO DO A MULTIPLE TABLE LOOK UP FOR Y GIVEN X AND Z
C      WITH EXTRAPOLATION PERMISSIBLE IN X DIR. FOR IEXX.LE.0
C      AND IN Z DIR. FOR IEXZ.LE.0
C                Y SQUARED               CONSIDERED FOR ISQ.EQ.1
C                LOG X AND LOG Z         CONSIDERED FOR ISQ.EQ.2
C                LOG X, LOG Y AND LOG Z  CONSIDERED FOR ISQ.GE.3
C                         NOTE:
C                FIRST DIMENSION OF YT ARRAY IS NDX
C
C
C             CODER:  R.E.CHUPP    DATE:  11/18/80
C -----------------------------------------------------------------
      SUBROUTINE MTABLU (XT,YT,ZT,X,Y,Z,NX,NZ,NDX,IEXX,IEXZ,ISQ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XT(1), YT(NDX,1), ZT(1)
      BSQRT(SDUM)=SQRT(SDUM)
      BLOG(FDUM)=DLOG(FDUM)
      BEXP(EDUM)=EXP(EDUM)
C
      IDIR=1
      IMIN=1
      IMAX=NX
      IF (XT(NX).GE.XT(1)) GO TO 10
      IDIR=-1
      IMIN=NX
      IMAX=1
10    AX=X
      IF (IEXX.GT.0.AND.AX.LT.XT(IMIN)) AX=XT(IMIN)
      IF (IEXX.GT.0.AND.AX.GT.XT(IMAX)) AX=XT(IMAX)
      DO 20 I=2,NX
```

```
      DIFX=XT(I)-AX
      IF (IDIR.GT.0.AND.DIFX.GE.0.0) GO TO 30
      IF (IDIR.LT.0.AND.DIFX.LE.0.0) GO TO 30
20    CONTINUE
      I=NX
30    I1=I-1
      IDIR=1
      IMIN=1
      IMAX=NZ
      IF (ZT(NZ).GE.ZT(1)) GO TO 40
      IDIR=-1
      IMIN=NZ
      IMAX=1
40    AZ=Z
      IF (IEXZ.GT.0.AND.AZ.LT.ZT(IMIN)) AZ=ZT(IMIN)
      IF (IEXZ.GT.0.AND.AZ.GT.ZT(IMAX)) AZ=ZT(IMAX)
      DO 50 J=2,NZ
      DIFZ=ZT(J)-AZ
      IF (IDIR.GT.0.AND.DIFZ.GE.0.0) GO TO 60
      IF (IDIR.LT.0.AND.DIFZ.LE.0.0) GO TO 60
50    CONTINUE
      J=NZ
60    J1=J-1
      Y11=YT(I1,J1)
      Y21=YT(I,J1)
      Y12=YT(I1,J)
      Y22=YT(I,J)
      X1=XT(I1)
      X2=XT(I)
      Z1=ZT(J1)
      Z2=ZT(J)
      IF (ISQ-1) 90,70,80
70    Y11=Y11**2
      Y21=Y21**2
      Y12=Y12**2
      Y22=Y22**2
      GO TO 90
80    X1=BLOG(X1)
      X2=BLOG(X2)
      Z1=BLOG(Z1)
      Z2=BLOG(Z2)
      DIFX=X2-BLOG(AX)
      DIFZ=Z2-BLOG(AZ)
      IF (ISQ.LE.2) GO TO 90
      Y11=BLOG(Y11)
      Y21=BLOG(Y21)
      Y12=BLOG(Y12)
      Y22=BLOG(Y22)
90    DUM=DIFX/(X2-X1)
      Y1=Y21-DUM*(Y21-Y11)
      Y2=Y22-DUM*(Y22-Y12)
      Y=Y2-DIFZ*(Y2-Y1)/(Z2-Z1)
      IF (ISQ.EQ.1.AND.Y.LE.0.0) Y=0.0
      IF (ISQ.EQ.1.AND.Y.GT.0.0) Y=BSQRT(Y)
      IF (ISQ.GE.3) Y=BEXP(Y)
      RETURN
      END
C -----------------------------------------------------------------
C
C                          SUBROUTINE     INCHEK
C
C       SUBROUTINE TO CHECK RANGE OF PARAMETER TO BE SURE THEY
C       ARE WITHIN THE APPROPRIATE RANGES FOR WHICH THE CORRELATIONS
```

```
C       FOR AMUL AND ALPHA ARE BASED
C
C ---------------------------------------------------------------------
      SUBROUTINE INCHEK (CL,AKT,AKP,AKH,ASH,ADTC,AKTH,ROUGH,TYPE,NOKNRC
     1,IERR,NOKNIF,DIRECT)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL OUT
      DIMENSION CL(30), AKT(30), AKP(30), AKH(30), ASH(30), ADTC(30),
     1 AKTH(30), ROUGH(30), VMIN(4), VMAX(4), PMIN(6,2), PMAX(6,2)

      CHARACTER STEP*5, STRGHT*5, TMIX*5, ALTSD*5, BLNK*6, STAR*6
      CHARACTER TYPE*5, TYP*6   , DOWN*6,    UP*6, DIRECT*4

      DATA STEP/'STEP '/,STRGHT/'STRAI'/,TMIX/'MIXED'/
      DATA ALTSD/'LTSD'/
      DATA BLNK/'      '/,STAR/' **** '/

      DATA VMIN/30.,0.,.54,0./
      DATA VMAX/90.,3.3,4.,27000./
      DATA PMIN/.21,5.1,.09,.85,1.16,2.02,.5,5.1,.35,4.1,1.02,4./
      DATA PMAX/2.6,29.4,1.,40.,1.76,29.4,1.5,28.,.5,19.4,1.9,12.5/
      BABS(ADUM)=ABS(ADUM)
      BMAX1(ARG1,ARG2)=DMAX1(ARG1,ARG2)
C
C     CHECK INPUT DATA RANGE
C
      TYP=TYPE
      IERR=0
      OUT=.FALSE.
      WRITE (6,10)
10    FORMAT(' ',T20,'I N P U T   D A T A   R A N G E   C H E C K')
      DO 410 J=1,NOKNRC
      IF (NOKNRC.LE.1) WRITE (6,20)
20    FORMAT(' ALL KNIVES')
      IF (NOKNRC.GT.1) WRITE (6,30) J
30    FORMAT(' KNIFE',I3)
      WRITE (6,40)
40    FORMAT(T13,'VARIABLE',T28,'MIN',T36,'VALUE',T46,'MAX')
      IF (TYPE.NE.TMIX) GO TO 50
      TYP=STRGHT
      IF (BABS(ASH(J)).GT.CL(J)) TYP=STEP
C                                     BOTH STRAIGHT AND STEP SEALS
C                                  CHECK THETA
50    DOWN=BLNK
      UP=BLNK
      IF (AKTH(J).GE.VMIN(1)) GO TO 60
      DOWN=STAR
      OUT=.TRUE.
60    IF (AKTH(J).LE.VMAX(1)) GO TO 70
      UP=STAR
      OUT=.TRUE.
70    WRITE (6,80) DOWN,VMIN(1),AKTH(J),VMAX(1),UP
80    FORMAT (' ',T16,'THETA',A6,F4.0,F10.1,F8.0,A6)
      IF (TYP.EQ.STEP) GO TO 150
C                                                    STRAIGHT SEALS
C                                  CHECK KT/CL
      DOWN=BLNK
      UP=BLNK
      VAL=AKT(J)/CL(J)
      IF (VAL.GE.VMIN(2)) GO TO 90
      DOWN=STAR
      OUT=.TRUE.
90    IF (VAL.LE.VMAX(2)) GO TO 100
```

```
      UP=STAR
      OUT=.TRUE.
100   WRITE (6,110) DOWN,VMIN(2),VAL,VMAX(2),UP
110   FORMAT (T16,'KT/CL',A6,F4.1,F10.3,F8.1,A6)
      IF (NOKNRC.GT.1.AND.J.EQ.1) GO TO 370
      IF (J.EQ.NOKNIF) GO TO 370
C                                   CHECK (KP-KT)/KH
      DOWN=BLNK
      UP=BLNK
      VAL=(AKP(J+1)-(AKT(J)+AKT(J+1))/2.)/AKH(J+1)
      IF (VAL.GE.VMIN(3)) GO TO 120
      DOWN=STAR
      OUT=.TRUE.
120   IF (VAL.LE.VMAX(3)) GO TO 130
      UP=STAR
      OUT=.TRUE.
130   WRITE (6,140) DOWN,VMIN(3),VAL,VMAX(3),UP
140   FORMAT (T11,'(KP-KT)/KH',A6,F4.2,F10.3,F8.0,A6)
      GO TO 370
C                                                       STEPPED SEALS
150   JD=1
      IF (DIRECT.EQ.ALTSD) JD=2
      IF (TYPE.EQ.TMIX.AND.ASH(J).LT.0.) JD=2
      TSH=BABS(ASH(J))
      TKP=AKP(J)
      TKT=AKT(J)
      TKH=AKH(J)
      TDTC=ADTC(J)
      IF (J.GT.1.OR.NOKNRC.GT.1) GO TO 160
      IF (ASH(1).EQ.0..AND.ASH(2).NE.0.) TSH=BABS(ASH(2))
      IF (AKP(1).EQ.0..AND.AKP(2).NE.0.) TKP=AKP(2)
      IF (ADTC(1).EQ.0..AND.ADTC(2).NE.0.) TDTC=ADTC(2)
C                                   CHECK KT/CL
160   DOWN=BLNK
      UP=BLNK
      VAL=AKT(J)/CL(J)
      IF (VAL.GE.PMIN(1,JD)) GO TO 170
      TKT=CL(J)*PMIN(1,JD)
      DOWN=STAR
      OUT=.TRUE.
170   IF (VAL.LE.PMAX(1,JD)) GO TO 180
      UP=STAR
      OUT=.TRUE.
180   WRITE (6,190) DOWN,PMIN(1,JD),VAL,PMAX(1,JD),UP
190   FORMAT (T16,'KT/CL',A6,F4.2,F10.3,F8.1,A6)
      IF (DOWN.EQ.STAR) WRITE (6,200) TKT
200   FORMAT (T60,'FOR CALCULATION, KT RESET TO',F10.4)
C                                   CHECK KH/CL
      DOWN=BLNK
      UP=BLNK
      VAL=AKH(J)/CL(J)
      IF (VAL.GE.PMIN(2,JD)) GO TO 210
      TKH=CL(J)*PMIN(2,JD)
      DOWN=STAR
      OUT=.TRUE.
210   IF (VAL.LE.PMAX(2,JD)) GO TO 220
      UP=STAR
      OUT=.TRUE.
220   WRITE (6,230) DOWN,PMIN(2,JD),VAL,PMAX(2,JD),UP
230   FORMAT (T16,'KH/CL',A6,F4.1,F10.3,F8.1,A6)
      IF (DOWN.EQ.STAR) WRITE (6,240) TKH
240   FORMAT (T60,'FOR CALCULATION, KH RESET TO',F10.4)
      IF (J.EQ.1.AND.TSH.EQ.0.) GO TO 370
```

```
C                                       CHECK DTC/(KP-KT)
      DOWN=BLNK
      UP=BLNK
      VAL=TDTC/(TKP-TKT)
      IF (VAL.GE.PMIN(3,JD)) GO TO 250
      DOWN=STAR
      OUT=.TRUE.
250   IF (VAL.LE.PMAX(3,JD)) GO TO 260
      UP=STAR
      OUT=.TRUE.
260   WRITE (6,270) DOWN,PMIN(3,JD),VAL,PMAX(3,JD),UP
270   FORMAT (T10,'DTC/(KP-KT)',A6,F4.2,F10.3,F8.1,A6)
C                                       CHECK DTC/CL
      DOWN=BLNK
      UP=BLNK
      VAL=TDTC/CL(J)
      IF (VAL.GE.PMIN(4,JD)) GO TO 280
      DOWN=STAR
      OUT=.TRUE.
280   IF (VAL.LE.PMAX(4,JD)) GO TO 290
      UP=STAR
      OUT=.TRUE.
290   WRITE (6,300) DOWN,PMIN(4,JD),VAL,PMAX(4,JD),UP
300   FORMAT (T15,'DTC/CL',A6,F4.2,F10.3,F8.1,A6)
C                                       CHECK (KP-KT)/KH
      DOWN=BLNK
      UP=BLNK
      VAL=(TKP-TKT)/TKH
      IF (VAL.GE.PMIN(5,JD)) GO TO 310
      DOWN=STAR
      OUT=.TRUE.
310   IF (VAL.LE.PMAX(5,JD)) GO TO 320
      TKH1=(TKP-TKT)/PMAX(5,JD)
      TKH=BMAX1(TKH,TKH1)
      UP=STAR
      OUT=.TRUE.
320   WRITE (6,330) DOWN,PMIN(5,JD),VAL,PMAX(5,JD),UP
330   FORMAT (T11,'(KP-KT)/KH',A6,F4.2,F10.3,F8.2,A6)
      IF (UP.EQ.STAR) WRITE (6,240) TKH
C                                       CHECK SH/CL
      DOWN=BLNK
      UP=BLNK
      VAL=TSH/CL(J)
      IF (VAL.GE.PMIN(6,JD)) GO TO 340
      DOWN=STAR
      OUT=.TRUE.
340   IF (VAL.LE.PMAX(6,JD)) GO TO 350
      UP=STAR
      OUT=.TRUE.
350   WRITE (6,360) DOWN,PMIN(6,JD),VAL,PMAX(6,JD),UP
360   FORMAT (T16,'SH/CL',A6,F4.2,F10.3,F8.1,A6)
C                                     ROUGHNESS FOR STRAIGHT OR STEP SEALS
370   IF (ROUGH(J).LE.30.) GO TO 410
      DOWN=BLNK
      UP=BLNK
      VAL=(ROUGH(J)-30.)/(2.*CL(J))
      IF (VAL.GE.VMIN(4)) GO TO 380
      DOWN=STAR
      OUT=.TRUE.
380   IF (VAL.LE.VMAX(4)) GO TO 390
      UP=STAR
      OUT=.TRUE.
390   WRITE (6,400) DOWN,VMIN(4),VAL,VMAX(4),UP
```

```
400   FORMAT (T13,'(E-30)/D',A6,F4.1,F10.1,F8.0,A6)
410   CONTINUE
      IF (.NOT.OUT) GO TO 430
      WRITE (6,420)
420   FORMAT (' ',T10,13('----')/' ',T10,'W A R N I N G   SEAL CALCULATI
     1ON MAY BE IN ERROR'/T10,'**** INDICATES VARIABLES OUTSIDE RANGE OF
     2 DATA BASE'/T10,'USED FOR EMPIRICAL CORRELATION'/' ',T10,13('----'
     3))
      IERR=1
430   RETURN
      END
```

Program Listing

File: K21.FOR

```
C -----------------------------------------------------------------
C     File:    K21.FOR
C     Program: KTK
C     Version: 2
C
C     Revision History:
C     (1) Modified December 1993 to compile under OS/2 using the
C     WATCOM F77/386 V9.0 compiler.
C     (2) Modified June 1994 to change the input formats for
C     integration with the CFD user interface shell.
C -----------------------------------------------------------------
      SUBROUTINE GAMCAL (TR,PO,GAMMA)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     ------------------------------------------------------------------
C
C                          SUBROUTINE     GAMCAL
C
C          SUBROUTINE TO CALCULATE GAMMA FROM CURVE FIT EQUATIONS
C          WHERE GAMMA IS THE SPECIFIC HEAT RATIO FOR AIR
C
C          TEMPERATURE (TR) IN DEGREES RANKINE
C
C
C     ------------------------------------------------------------------
C
      TO=TR-459.69
C
C
C             INTERPOLATION FOR GAMMA                                * * *
C             GAMMA DOES NOT = F(P)  FOR T   GREATER THAN 1880 F
C             AK1 FOR 1 ATM,  AK2 FOR 10 ATM, AND AK3 FOR 40 ATM
      TO2=TO**2
      TO3=TO2*TO
      TO4=TO3*TO
      IF (TO-1880.) 20,10,10
10    GAMMA=1.354153-.1651369E-4*TO-.2495806E-8*TO2
      GO TO 30
20    AK1=1.405482-.3274806E-4*TO-.3159632E-7*TO2+.1272061E-10*TO3
      AK2=-AK1+1.425199-.8470722E-4*TO+.1401141E-7*TO2
      AK3=-AK1+1.491990-.2887446E-3*TO+.2682448E-6*TO2-.1419973E-9*TO3
     1+.2909987E-13*TO4
      A=(9.*AK3-39.*AK2)/10530.
      X=PO/14.696-1.
      GAMMA=A*X**2+(AK2/9.-A*9.)*X+AK1
30    CONTINUE
      RETURN
      END
C     DATA SET SFCDCOPTSB AT LEVEL 004 AS OF 08/12/83
      SUBROUTINE OPTSUB
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL OUT,ONEPT,PRNT
      CHARACTER*5 AOPTD,XTYPE, AOPT
      CHARACTER*4 TITLE, TITLEC, B

      COMMON /C13R/  PARVAL,AMIN,ROGCN
      COMMON /C13RA/ TITLE(20),TITLEC(20),AOPTD
      COMMON /C14R/  G,GP1,GP1GM1,GP1OG,GP1O2,GM1O2,XK(100)
      COMMON /C1234R/ PT(100),TTSR(100)
      COMMON /VECTR/ CL(30),AKR(30),AKT(30),AKP(30),AKH(30),ASH(30),ADTC
     1(30),AKTH(30),AKBETA(30),DIA(30),ROUGH(30),TR(30),A4FLOD(30),DELC
     2(30),DELE(30),KCCO(30),KECO(30),PKR(30)
      COMMON /OPTION/ IFIX(6),IOFFC(25), IDOMAN,IDAVLM,IDAVE,ISRCHP
```

```
     1,IPASSP
      COMMON /OPTL/ ONEPT

      COMMON /LIMIT/ XVMIN(25),XVMAX(25)
      COMMON /KNIFE/ XLMAX,HMAX,KO,IO,JO
      COMMON /OPTDAT/ ALENGH,DMIN,DMAX,ALPHA,PRATIO,XMW,ARFACT,NOW
     1,ICURVE,ICVTYP,IPRINT
      COMMON /OPTX  / AOPT
      COMMON /OPTMUM/ FMIN
      COMMON /BIND/ B(25)
C
      DIMENSION  XI(6), XMIN(6), GI(6), WOPT(2,2,20),OKT(2,2,20),
     1          OKP(2,2,20), OKH(2,2,20), OSH(2,2,20), OKTH(2,2,20),
     2          ORUF(2,2,20), VMIN(9), VMAX(9), STEPMN(6,2), STEPMX(6,2)

      CHARACTER    WORD(14)*8, TYP(2)*8, DIR(3)*4, EMSG(4)*8
      CHARACTER*5  ALTSD,STLD, STEP, STRGHT
      CHARACTER*4  BOTH, EMIN,EMAX
      CHARACTER    WRD1*8, WRD2*4, WRD3*4
      CHARACTER    OTYPE*5, ODIR*5, TYPE*5, DIRECT*5
C
      DATA ALTSD/'LTSD '/,STLD/'STLD '/
      DATA STEP/'STEP '/,STRGHT/'STRAI'/,BOTH/'BOTH'/
      DATA TYP/'STRAIGHT',' STEPPED'/
      DATA WORD/'KT','KP','KH/KP','SH','KTHETA','ROUGH','KN','END','KT/C
     1L','(KP-KT)/','KH/CL','DTC/(KP-','DTC/CL','SH/CL'/
      DATA DIR/'STLD','LTSD','    '/
      DATA OUT/.FALSE./,EMIN/' MIN'/,EMAX/' MAX'/
      DATA EMSG/' IS OUTS','IDE EMPI','RICAL DA','TA RANGE'/

      DATA VMIN/.009,.010,1.,.010,30.,30.,.001,.54,0./
      DATA VMAX/1.,1.,1.,1.,90.,300.,3.3,4.,27000./
      DATA STEPMN/.21,5.1,.09,.85,1.16,2.02,.5,5.1,.35,4.1,1.02,4./
      DATA STEPMX/2.6,29.4,1.,40.,1.76,29.4,1.5,28.,.5,19.4,1.9,12.5/
C
      BABS(ADUM)=ABS(ADUM)
      BMAX1(ARG1,ARG2)=DMAX1(ARG1,ARG2)
      BMIN1(ARG1,ARG2)=DMIN1(ARG1,ARG2)
C
C                        READ OPTIMIZATION INPUT DATA
C
      WRITE (6,10)
10    FORMAT(/,4X,'K N I F E -- T O -- K N I F E   S E A L   D E S I',
     1           ' G N   M O D E L')
      WRITE (6,20) TITLE,TITLEC
20    FORMAT(/,4X,20A4,/,4X,20A4)
      IDAVLM=100
      ONEPT=.FALSE.
      PRNT=.FALSE.
C
C ****************************************************************
C RECORD 7 FORMAT WAS CHANGED TO ALLOW LARGER WIDTH FOR FLOATING
C POINT NUMBERS
C ****************************************************************
      READ(5,30,END=1400) XLMAX,HMAX,OTYPE,ODIR,ISRCHP,IPASSP,IWPRNT
30    FORMAT (2E12.4E3,2A5,3I5)
C ****************************************************************
C
      IF (IWPRNT.GT.0) PRNT=.TRUE.
      WRITE (6,40) XLMAX,HMAX,PRATIO,OTYPE,ODIR,ISRCHP,IPASSP,IWPRNT
40    FORMAT(' LMAX',G14.6,'   HMAX',G14.6,'   PRATIO',G14.6,'   TYPE '
     1,A5,'   DIRECTION ',A5,'   ISRCHP',I2,'   IPASSP',I2,'   IWPRNT',
     2I2)
```

```
C
50    CONTINUE
C
C ****************************************************************
C RECORDS 8 AND 9 FORMAT WAS CHANGED TO ALLOW LARGER WIDTH FOR
C FLOATING POINT NUMBERS
C ****************************************************************
      READ(5,60,END=1400) WRD1,WRD2,WRD3,D1,D2
60    FORMAT (A8, A4, A4, 4X, 2E12.4E3)
C ****************************************************************
C
      IF (WRD1.EQ.WORD(8)) GO TO 480
      IF (WRD1.EQ.WORD(1)) GO TO 90
      IF (WRD1.EQ.WORD(2)) GO TO 100
      IF (WRD1.EQ.WORD(3)) GO TO 110
      IF (WRD1.EQ.WORD(4)) GO TO 120
      IF (WRD1.EQ.WORD(5)) GO TO 130
      IF (WRD1.EQ.WORD(6)) GO TO 160
      IF (WRD1.EQ.WORD(7)) GO TO 190
      IF (WRD1.EQ.WORD(9)) GO TO 200
      IF (WRD1.EQ.WORD(10)) GO TO 260
      IF (WRD1.EQ.WORD(11)) GO TO 320
      IF (WRD1.EQ.WORD(12)) GO TO 360
      IF (WRD1.EQ.WORD(13)) GO TO 400
      IF (WRD1.EQ.WORD(14)) GO TO 440
70    WRITE (6,80) WRD1,WRD2,WRD3,D1,D2
80    FORMAT (' OPT LIMIT CARD---',A8,2A4,2G14.6,' COULD NOT BE INTERPRE
     1TED'/' DEFAULT VALUES LEFT IN EFFECT')
      GO TO 50
C                                          KT
90    IF (D1.GT.0.) VMIN(1)=D1
      IF (D2.GT.0.) VMAX(1)=D2
      GO TO 50
C                                          KP
100   IF (D1.GT.0.) VMIN(2)=D1
      IF (D2.GT.0.) VMAX(2)=D2
      GO TO 50
C                                          KH/KP
110   IF (D1.GT.0.) VMIN(3)=D1
      IF (D2.GT.0.) VMAX(3)=D2
      GO TO 50
C                                          SH
120   IF (D1.GT.0.) VMIN(4)=D1
      IF (D2.GT.0.) VMAX(4)=D2
      GO TO 50
C                                          KTHETA
130   IF (D1.LE.0.) GO TO 150
      IF (D1.LT.VMIN(5)) WRITE (6,140) EMIN,EMSG
140   FORMAT (' INPUT KTHETA',A4,4A8)
      VMIN(5)=D1
150   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.VMAX(5)) WRITE (6,140) EMAX,EMSG
      VMAX(5)=D2
      GO TO 50
C                                     ROUGHNESS
160   IF (D1.LE.0.) GO TO 170
      VMIN(6)=BMAX1(VMIN(6),D1)
170   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.VMAX(6)) WRITE (6,180) EMAX,EMSG
180   FORMAT (' INPUT ROUGH',A4,4A8)
      VMAX(6)=D2
      GO TO 50
C                                       KN
```

```
190   KN1=D1
      KN2=D2
      GO TO 50
C                                          KT/CL
200   IF (WRD3.NE.DIR(3)) GO TO 230
      IF (D1.LE.0.) GO TO 220
      IF (D1.LT.VMIN(7)) WRITE (6,210) EMIN,EMSG
210   FORMAT (' INPUT KT/CL',A4,4A8)
      VMIN(7)=D1
220   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.VMAX(7)) WRITE (6,210) EMAX,EMSG
      VMAX(7)=D2
      GO TO 50
230   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 240
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
240   IF (D1.LE.0.) GO TO 250
      IF (D1.LT.STEPMN(1,JD)) WRITE (6,210) EMIN,EMSG
      STEPMN(1,JD)=D1
250   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(1,JD)) WRITE (6,210) EMAX,EMSG
      STEPMX(1,JD)=D2
      GO TO 50
C                                          (KP-KT)/KH
260   IF (WRD3.NE.DIR(3)) GO TO 290
      IF (D1.LE.0.) GO TO 280
      IF (D1.LT.VMIN(8)) WRITE (6,270) EMIN,EMSG
270   FORMAT (' INPUT (KP-KT)/KH',A4,4A8)
      VMIN(8)=D1
280   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.VMAX(8)) WRITE (6,270) EMAX,EMSG
      VMAX(8)=D2
      GO TO 50
290   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 300
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
300   IF (D1.LE.0.) GO TO 310
      IF (D1.LT.STEPMN(5,JD)) WRITE (6,270) EMIN,EMSG
      STEPMN(5,JD)=D1
310   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(5,JD)) WRITE (6,270) EMAX,EMSG
      STEPMX(5,JD)=D2
      GO TO 50
C                                          KH/CL
320   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 330
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
330   IF (D1.LE.0.) GO TO 350
      IF (D1.LT.STEPMN(2,JD)) WRITE (6,340) EMIN,EMSG
340   FORMAT (' INPUT KH/CL',A4,4A8)
      STEPMN(2,JD)=D1
350   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(2,JD)) WRITE (6,340) EMAX,EMSG
      STEPMX(2,JD)=D2
      GO TO 50
C                                          DTC/(KP-KT)
360   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 370
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
```

```
370   IF (D1.LE.0.) GO TO 390
      IF (D1.LT.STEPMN(3,JD)) WRITE (6,380) EMIN,EMSG
380   FORMAT (' INPUT DTC/(KP-KT)',A4,4A8)
      STEPMN(3,JD)=D1
390   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(3,JD)) WRITE (6,380) EMAX,EMSG
      STEPMX(3,JD)=D2
      GO TO 50
C                                        DTC/CL
400   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 410
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
410   IF (D1.LE.0.) GO TO 430
      IF (D1.LT.STEPMN(4,JD)) WRITE (6,420) EMIN,EMSG
420   FORMAT (' INPUT DTC/CL',A4,4A8)
      STEPMN(4,JD)=D1
430   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(4,JD)) WRITE (6,420) EMAX,EMSG
      STEPMX(4,JD)=D2
      GO TO 50
C                                        SH/CL
440   JD=1
      IF (WRD3.EQ.DIR(1)) GO TO 450
      IF (WRD3.NE.DIR(2)) GO TO 70
      JD=2
450   IF (D1.LE.0.) GO TO 470
      IF (D1.LT.STEPMN(6,JD)) WRITE (6,460) EMIN,EMSG
460   FORMAT (' INPUT SH/CL',A4,4A8)
      STEPMN(6,JD)=D1
470   IF (D2.LE.0.) GO TO 50
      IF (D2.GT.STEPMX(6,JD)) WRITE (6,460) EMAX,EMSG
      STEPMX(6,JD)=D2
      GO TO 50
480   CONTINUE
C                              CHECK LIMITS AGAINST DATA RANGE
      IF (OTYPE.NE.STRGHT.AND.OTYPE.NE.BOTH) GO TO 550
C                                   KT/CL  STRAIGHT
      TRY=VMIN(1)/CL(1)
      TRY1=VMAX(1)/CL(1)
      IF (TRY.GE.VMIN(7).AND.TRY1.LE.VMAX(7)) GO TO 500
      OUT=.TRUE.
      WRITE (6,490) DIR(3),VMIN(1),VMAX(1),VMIN(7),VMAX(7),TRY,TRY1
490   FORMAT (' KT LIMITS OUT OF DATA RANGE FOR KT/CL ',A4,T60,'KT LIM',
     12F8.3,' DATA RANGE',2F8.3/T82,' CALC RANGE',2G13.6)
500   IF (OUT) GO TO 510
      TRY=AKT(1)/CL(1)
      IF (TRY.LT.VMIN(7)) AKT(1)=CL(1)*VMIN(7)*1.02
      IF (TRY.GT.VMAX(7)) AKT(1)=CL(1)*VMAX(7)*.98
C                                  (KP-KT)/KH   STRAIGHT
510   TKH=VMIN(2)*VMIN(3)
      TKH1=VMAX(2)*VMAX(3)
      TRY=(VMIN(2)-VMAX(1))/TKH1
      TRY1=(VMAX(2)-VMIN(1))/TKH
      IF (TRY.GE.VMIN(8).AND.TRY1.LE.VMAX(8)) GO TO 530
      OUT=.TRUE.
      WRITE (6,520) DIR(3),VMIN(2),VMAX(2),VMIN(8),VMAX(8),TKH,TKH1,TRY
     1,TRY1
520   FORMAT (' KP,KT,KH LIMITS OUT OF DATA RANGE FOR (KP-KT)/KH ',A4,T6
     10,'KP LIM',2F8.3,' DATA RANGE',2F8.3/T60,'KH LIM',2F8.3,' CALC RAN
     2GE',2G13.6)
530   IF (OUT) GO TO 550
      TRY=AKH(1)/AKP(1)
```

```
      IF (TRY.LT.VMIN(3)) AKH(1)=AKP(1)*VMIN(3)*1.02
      IF (TRY.GT.VMAX(3)) AKH(1)=AKP(1)*VMAX(3)*.98
      TRY=(AKP(1)-AKT(1))/AKH(1)
      IF (TRY.GE.VMIN(8)) GO TO 540
      AKP(1)=AKT(1)/(1.-VMIN(3)*VMIN(8))*1.02
      AKH(1)=AKP(1)*VMIN(3)*1.02
540   IF (TRY.LE.VMAX(8)) GO TO 550
      AKP(1)=AKT(1)/(1.-VMAX(3)*VMAX(8))*.98
      AKH(1)=AKP(1)*VMAX(3)*.98
550   IF (OTYPE.NE.STEP.AND.OTYPE.NE.BOTH) GO TO 780
C                                  KT/CL  (STEP)
      DO 560 JD=1,2
      IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 560
      TRY=VMIN(1)/CL(1)
      TRY1=VMAX(1)/CL(1)
      IF (TRY.LE.STEPMX(1,JD).AND.TRY1.GE.STEPMN(1,JD)) GO TO 560
      OUT=.TRUE.
      WRITE (6,490) DIR(JD),VMIN(1),VMAX(1),STEPMN(1,JD),STEPMX(1,JD)
     1,TRY,TRY1
560   CONTINUE
      IF (OUT) GO TO 580
      DO 570 JD=1,2
      IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 570
      TRY=AKT(1)/CL(1)
      IF (TRY.LT.STEPMN(1,JD)) AKT(1)=CL(1)*STEPMN(1,JD)*1.02
      IF (TRY.GT.STEPMX(1,JD)) AKT(1)=CL(1)*STEPMX(1,JD)*.98
570   CONTINUE
C                                  DTC/CL
580   IF (ADTC(1).GT.0.) GO TO 600
      OUT=.TRUE.
      WRITE (6,590)
590   FORMAT (' DTC NOT DEFINED FOT STEP SEAL')
600   TRY=ADTC(1)/CL(1)
      DO 620 JD=1,2
      IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 620
      IF (TRY.GE.STEPMN(4,JD).AND.TRY.LE.STEPMX(4,JD)) GO TO 620
      OUT=.TRUE.
      WRITE (6,610) DIR(JD),STEPMN(4,JD),STEPMX(4,JD),TRY
610   FORMAT (' DTC,CL INPUT IS OUT OF DATA RANGE FOR DTC/CL ',A4,T60,'
     1DATA RANGE',2F8.3,' CALC RANGE',2G13.6)
620   CONTINUE
C                                       DTC/(KP-KT)
      TRY=ADTC(1)/(VMIN(2)-VMIN(1))
      TRY1=ADTC(1)/(VMAX(2)-VMIN(1))
      DO 640 JD=1,2
      IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 640
      IF (TRY1.GE.STEPMN(3,JD).AND.TRY1.LE.STEPMX(3,JD)) GO TO 640
      OUT=.TRUE.
      WRITE (6,630) DIR(JD),VMIN(2),VMAX(2),STEPMN(3,JD),STEPMX(3,JD)
     1,TRY,TRY1
630   FORMAT (' KP,KT LIMITS OUT OF DATA RANGE FOR DTC/(KP-KT) ',A4,T60,
     1'KP LIM',2F8.3,' DATA RANGE',2F8.3/T82,' CALC RANGE',2G13.6)
640   CONTINUE
      IF (OUT) GO TO 660
      DO 650 JD=1,2
      IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 650
      TRY=ADTC(1)/(AKP(1)-AKT(1))
      IF (TRY.LT.STEPMN(3,JD)) AKP(1)=(ADTC(1)+AKT(1)*STEPMN(3,JD))
     1/STEPMN(3,JD)*1.02
      IF (TRY.GT.STEPMX(3,JD)) AKP(1)=(ADTC(1)+AKT(1)*STEPMX(3,JD))
     1/STEPMX(3,JD)*.98
650   CONTINUE
C                                       SH/CL
```

```
660    TRY=VMIN(4)/CL(1)
       TRY1=VMAX(4)/CL(1)
       DO 680 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 680
       IF (TRY.GE.STEPMN(6,JD).AND.TRY1.LE.STEPMX(6,JD)) GO TO 680
       OUT=.TRUE.
       WRITE (6,670) DIR(JD),VMIN(4),VMAX(4),STEPMN(6,JD),STEPMX(6,JD)
      1,TRY,TRY1
670    FORMAT (' SH LIMITS OUT OF DATA RANGE FOR SH/CL ',A4,T60,'SH LIM',
      12F8.3,' DATA RANGE',2F8.3/T82,' CALC RANGE',2G13.6)
680    CONTINUE
       IF (OUT) GO TO 700
       DO 690 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 690
       TRY=ASH(1)/CL(1)
       IF (TRY.LT.STEPMN(6,JD)) ASH(1)=CL(1)*STEPMN(6,JD)*1.02
       IF (TRY.GT.STEPMX(6,JD)) ASH(1)=CL(1)*STEPMX(6,JD)*.98
690    CONTINUE
C                                    (KP-KT)/KH   STEP
700    TKH=VMIN(2)*VMIN(3)
       TKH1=VMAX(2)*VMAX(3)
       TRY=(VMIN(2)-VMIN(1))/TKH1
       TRY1=(VMAX(2)-VMIN(1))/TKH
       DO 710 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 710
       IF (TRY.GE.STEPMN(5,JD).AND.TRY1.LE.STEPMX(5,JD)) GO TO 710
       OUT=.TRUE.
       WRITE (6,520) DIR(JD),VMIN(2),VMAX(2),STEPMN(5,JD),STEPMX(5,JD)
      1,TRY,TRY1,TKH,TKH1
710    CONTINUE
       IF (OUT) GO TO 740
       TRY=AKH(1)/AKP(1)
       IF (TRY.LT.VMIN(3)) AKH(1)=AKP(1)*VMIN(3)*1.02
       IF (TRY.GT.VMAX(3)) AKH(1)=AKP(1)*VMAX(3)*.98
       DO 730 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 730
       TRY=(AKP(1)-AKT(1))/AKH(1)
       IF (TRY.GE.STEPMN(5,JD)) GO TO 720
       AKP(1)=AKT(1)/(1.-VMIN(3)*STEPMN(5,JD))*1.02
       AKH(1)=AKP(1)*VMIN(3)*1.02
720    IF (TRY.LE.STEPMX(5,JD)) GO TO 730
       AKP(1)=AKT(1)/(1.-VMAX(3)*STEPMX(5,JD))*.98
       AKH(1)=AKP(1)*VMAX(3)*.98
730    CONTINUE
C                                      KT/CL  STEP
740    TRY=VMIN(2)*VMIN(3)/CL(1)
       TRY1=VMAX(2)*VMAX(3)/CL(1)
       PL1=VMIN(2)*VMIN(3)
       PL2=VMAX(2)*VMAX(3)
       DO 760 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 760
       IF (TRY.GE.STEPMN(2,JD).AND.TRY1.LE.STEPMX(2,JD)) GO TO 760
       OUT=.TRUE.
       WRITE (6,750) DIR(JD),PL1,PL2,STEPMN(2,JD),STEPMX(2,JD),TRY,TRY1
750    FORMAT (' KH LIMITS OUT OF DATA RANGE FOR KH/CL ',A4,T60,'KH LIM',
      12F8.3,' DATA RANGE',2F8.3/T82,' CALC RANGE',2G13.6)
760    CONTINUE
       IF (OUT) GO TO 780
       DO 770 JD=1,2
       IF (ODIR.NE.DIR(JD).AND.ODIR.NE.BOTH) GO TO 770
       TRY=AKH(1)/CL(1)
       IF (TRY.LT.STEPMN(2,JD)) AKH(1)=CL(1)*STEPMN(2,JD)*1.02
       IF (TRY.GT.STEPMX(2,JD)) AKH(1)=CL(1)*STEPMX(2,JD)*.98
```

```
770   CONTINUE
780   CONTINUE
C
C        END  READING AND CHECKING INPUT DATA
C
      IF (KN1.LE.0) KN1=1
      IF (KN2.LE.0) KN2=7
      IF (KN2.GT.20) KN2=20
      NOKNIF=KN2
      TYPE=OTYPE
      IF (OTYPE.EQ.BOTH) TYPE=STEP
      DIRECT=ODIR
      IF (ODIR.EQ.BOTH) DIRECT=STLD
      DO 810 I=1,2
      DO 800 J=1,2
      DO 790 K=1,20
      WOPT(I,J,K)=0.
790   CONTINUE
800   CONTINUE
810   CONTINUE
C                          SET UP GEOMETRY DATA WHERE NOT READ IN
C
      IF (TR(1).LE.0.0) TR(1)=530.
      AVGDIA=BMAX1(DMIN,DMAX)
      IF (DIA(1).LE.0.0) DIA(1)=AVGDIA
      IF (KCCO(1).NE.0) GO TO 820
      KCCO(1)=-1
      IF (TYPE.EQ.STRGHT) KCCO(1)=1
820   IF (KECO(1).NE.0) GO TO 830
      KECO(1)=-1
      IF (TYPE.EQ.STRGHT) KECO(1)=1
830   IF (NOKNIF.LE.1) GO TO 880
      DUM=NOKNIF
      IF (TYPE.NE.STEP) GO TO 850
      DEL=ASH(1)*2.
      DEL=BABS(DEL)
      IF (DIRECT.EQ.ALTSD) DEL=-DEL
      IF (DIRECT.EQ.ALTSD) GO TO 840
C                                                    STLD DIRECTION
      IF (DMIN.GT.0.) DIA(1)=DMIN
      IF (DMAX.GT.0.) DIA(1)=DMAX-(DUM-1.)*DEL
      GO TO 850
C                                                    LTSD DIRECTION
840   IF (DMAX.GT.0.) DIA(1)=DMAX
      IF (DMIN.GT.0.) DIA(1)=DMIN-(DUM-1.)*DEL
850   SUM=DIA(1)
      PKR(1)=ARFACT*AKR(1)
      DO 870 I=2,NOKNIF
      IF (CL(I).LE.0.0) CL(I)=CL(1)
      IF (AKR(I).LE.0.0) AKR(I)=AKR(1)
      IF (AKT(I).LE.0.0) AKT(I)=AKT(1)
      IF (AKP(I).LE.0.0) AKP(I)=AKP(1)
      IF (AKH(I).LE.0.0) AKH(I)=AKH(1)
      IF (ASH(I).LE.0.0) ASH(I)=ASH(1)
      IF (ADTC(I).LE.0.0) ADTC(I)=ADTC(1)
      IF (AKTH(I).LE.0.0) AKTH(I)=AKTH(1)
      IF (AKBETA(I).LE.0.0) AKBETA(I)=AKBETA(1)
      IF (TR(I).LE.0.0) TR(I)=TR(1)
      IF (ROUGH(I).LE.0.0) ROUGH(I)=ROUGH(1)
      IF (A4FLOD(I).LE.0.0) A4FLOD(I)=A4FLOD(1)
      IF (KCCO(I).EQ.0) KCCO(I)=KCCO(1)
      IF (KECO(I).EQ.0) KECO(I)=KECO(1)
      IF (DIA(I).GT.0.0) GO TO 860
```

```
      IF (TYPE.NE.STEP) DIA(I)=DIA(I-1)
      IF (TYPE.EQ.STEP) DIA(I)=DIA(I-1)+DEL
860   SUM=SUM+DIA(I)
      PKR(I)=ARFACT*AKR(I)
870   CONTINUE
      AVGDIA=SUM/DUM
880   KCCO(1)=-1
      KECO(NOKNIF)=-1
C
C        PRINT INPUT INFORMATION
C
      WRITE (6,890)
890   FORMAT (' '/' ',T25,'INITIAL VALUES TO BEGIN OPTIMIZATION')
      I=1
      IF (TYPE.EQ.STEP) I=2
      WRITE (6,900) G,XMW,NOKNIF,TYP(I),DIRECT,ALENGH,AVGDIA,ALPHA,PT(1)
900   FORMAT (1H /1H ,'SPECIFIC HEAT RATIO   (GAMMA)  =',F10.4/1H ,'MOLE
     1CULAR WEIGHT               =',F10.4/1H0,'MAX NUMBER OF KNIVES
     2      =',I10/1H ,'SEAL TYPE                        =',2X,A8/1H ,'FLO
     3W DIRECTION                =',6X,A4/1H ,'SEAL LENGTH (2-D SEAL)
     4       =',F10.4,' (INCHES)'/1H ,'AVG. KNIFE DIAMETER (3-D SEAL) ='
     5,F10.4,' (INCHES)'/1H ,'FLOW DIVERGENCE ANGLE (ALPHA)  =',F10.4,'
     6(DEGREES)'/1H ,'INLET TOTAL PRESSURE           =',F10.4,' (PSIA)')
      WRITE (6,910)
910   FORMAT (1H ,4X, 'K N I F E   G E O M E T R Y   D A T A'/' KNIFE
     1 CL     KR      KT     KP     KH     SH     DTC  ','THETA  BETA
     2DIA   ROUGH  TEMP')
      WRITE (6,920)
920   FORMAT (1H ,'  NO.   (IN)   (IN)    (IN)   (IN)   (IN)   (IN)   (I
     1N)  ','(DEG) (DEG)  (IN)   (RMS) (DEGR)'/' -----  ------ ------- -
     2----- ------ ------ ------ ------ ','----- ----- ------  ----- ---
     3---'/)
      WRITE (6,930) (I,CL(I),PKR(I),AKT(I),AKP(I),AKH(I),ASH(I),ADTC(I)
     1,AKTH(I),AKBETA(I),DIA(I),ROUGH(I),TR(I),I=1,NOKNIF)
930   FORMAT (1H ,I4,F9.4,F8.5,5F7.4,2F6.1,F7.4,F7.2,F7.1)


      IF (OUT) WRITE (6,531)
 531  FORMAT (' OPTIMIZATION ABANDONED BECAUSE DATA OUT OF RANGE')
      IF (OUT) RETURN


      XI(1)=AKT(1)
      XI(2)=AKP(1)
      XI(3)=AKH(1)/AKP(1)
      XI(4)=ASH(1)
      XI(5)=BMIN1(AKTH(1),89.0D0)
      XI(6)=BMAX1(ROUGH(1),30.0D0)
      WMIN=100.
      DO 940 I=1,14
      IOFFC(I)=0
940   CONTINUE
      IOFFC(15)=1
      DO 950 I=1,8
      XVMIN(I)=VMIN(I)
      XVMAX(I)=VMAX(I)
950   CONTINUE
      XVMIN(15)=VMIN(9)
      XVMAX(15)=VMAX(9)
      DO 960 I=1,6
      IF (XVMAX(I).LE.XVMIN(I)) XI(I)=XVMIN(I)
      IFIX(I)=0
960   CONTINUE
```

```
      IF (OTYPE.NE.STRGHT.AND.OTYPE.NE.BOTH) GO TO 1090
C                                            OPTIMIZE STRAIGHT SEAL
      IO=1
      JO=1
      KO=KN1
      IFIX(4)=1
      IOFFC(7)=0
      IOFFC(8)=0
      IOFFC(9)=1
      IOFFC(10)=1
      IOFFC(11)=1
      IOFFC(12)=1
      IOFFC(13)=1
      IOFFC(14)=1
970   WRITE (6,980) KO
980   FORMAT ('
```

```
OPTIMIZATION STEP FOR STRAIGHT SEAL WITH NO. KNIVES =',I
     13)
      DUM=KO-1
      IF (XLMAX.GT.0..AND.KO.GT.1) XVMAX(2)=(XLMAX-XI(1))/DUM
      IF (XI(2).GE.XVMAX(2)) XI(2)=.98*XVMAX(2)

      WRITE(6,989)
  989 FORMAT(/,2X,'RANGES AVAILABLE FROM EMPIRICAL DATA',
     +       /,T10,'MIN',T24,'INITIAL',T38,'MAX')
      DO 3121 I = 1, 6
 3121 WRITE(6,990) I,XVMIN(I),XI(I),XVMAX(I),WORD(I)
  990 FORMAT(I5,3G14.6,6X,A8)

      WRITE(6,1000) XVMIN(7),XVMAX(7),WORD(9)
      WRITE(6,1001) XVMIN(8),XVMAX(8),WORD(10),WORD(11)
 1000 FORMAT('    7',G14.6,14X,G14.6,6X,A8)
 1001 FORMAT('    8',G14.6,14X,G14.6,6X,A8,A8)

      KCCO(1)=0
      KECO(1)=0
      CALL VMM (XI,6,XMIN,PRNT)
      CALL FUNCT (F,GI,XMIN,1,0,.TRUE.)
      WRITE (6,1010) TYP(1),KO,FMIN
1010  FORMAT (' PARAMETER VALUES AND DERIVATIVES FOR',5X,A8,' SEAL',I6,'
     1 KNIVES',5X,'MIN FLOW',G14.6)
      WRITE (6,1020)
1020  FORMAT (' ',T30,'MIN',T44,'OPTIMUM',T58,'MAX',T72,'DEL W/DEL X')
      WRITE (6,1030) (B(I),XVMIN(I),XMIN(I),XVMAX(I),GI(I),I=1,6)
1030  FORMAT (T18,A1,T20,'KT',T26,4G14.6/T18,A1,T20,'KP',T26,4G14.6/T18,
     1A1,T20,'KH/KP',T26,4G14.6/T18,A1,T20,'SH',T26,4G14.6/T18,A1,T20,'K
     2THETA',T26,4G14.6/T18,A1,T20,'ROUGH',T26,4G14.6)
      T1=XMIN(1)/CL(1)
      T2=(XMIN(2)-XMIN(1))/(XMIN(2)*XMIN(3))
      WRITE (6,1040)
1040  FORMAT (' CONSTRAINTS',T30,'MIN',T44,'VALUE',T58,'MAX')
      WRITE (6,1050) B(7),XVMIN(7),T1,XVMAX(7),B(8),XVMIN(8),T2,XVMAX(8)
1050  FORMAT (T14,A1,T16,'KT/CL',T26,3G14.6/T14,A1,T16,'(KP-KT)/KH',T26,
     13G14.6)
      WRITE (6,1060)
1060  FORMAT (' ',T20,'*  INDICATES BINDING CONSTRAINTS')
      WOPT(IO,JO,KO)=FMIN
      OKT(IO,JO,KO)=XMIN(1)
      OKP(IO,JO,KO)=XMIN(2)
      OKH(IO,JO,KO)=XMIN(2)*XMIN(3)
      OSH(IO,JO,KO)=XMIN(4)
      OKTH(IO,JO,KO)=XMIN(5)
      ORUF(IO,JO,KO)=XMIN(6)
      IF (FMIN.GE.WMIN) GO TO 1070
      WMIN=FMIN
      IOP=IO
      JOP=JO
      KOP=KO
1070  DO 1080 I=1,6
      XI(I)=XMIN(I)
1080  CONTINUE
      KO=KO+1
      IF (KO.LE.KN2) GO TO 970
1090  IF (OTYPE.NE.STEP.AND.OTYPE.NE.BOTH) GO TO 1270
C                                          OPTIMIZE STEPPED  SEAL
      IO=2
      KO=KN1
      IFIX(4)=0
      IOFFC(4)=0
```

```
      IOFFC(7)=1
      IOFFC(8)=1
      IOFFC(9)=0
      IOFFC(10)=0
      IOFFC(11)=0
      IOFFC(12)=0
      IOFFC(13)=0
      IOFFC(14)=0
1100  IF (ODIR.NE.STLD.AND.ODIR.NE.BOTH) GO TO 1200
C                                                  STLD DIRECTION
      DO 1110 I=1,6
      XVMIN(I+8)=STEPMN(I,1)
      XVMAX(I+8)=STEPMX(I,1)
1110  CONTINUE
      WRITE (6,1120) KO
1120  FORMAT ('
```

```
OPTIMIZATION STEP FOR STEPPED SEAL, STLD DIRECTION, WITH
     1 NO. KNIVES =',I3)
      DUM=KO-1
      TKH=XI(2)*XI(3)
      IF (TKH.GE.HMAX) XI(2)=.5*HMAX/XI(3)
      IF (XLMAX.GT.0..AND.KO.GT.1) XVMAX(2)=(XLMAX-XI(1))/DUM
      IF (XI(2).GE.XVMAX(2)) XI(2)=.98*XVMAX(2)
      IF (HMAX.GT.0..AND.KO.GT.1) XVMAX(4)=(HMAX-XI(2)*XI(3))/DUM
      IF (XI(4).GE.XVMAX(4)) XI(4)=.98*XVMAX(4)
      WRITE (6,990) (I,XVMIN(I),XI(I),XVMAX(I),I=1,6)
      WRITE (6,1000) (I,XVMIN(I),XVMAX(I),I=9,14)
      TRY=ADTC(1)/(XVMAX(2)-XVMIN(1))
      IF (TRY.LE.XVMAX(11)) GO TO 1140
      WRITE (6,1130)
1130  FORMAT (' MAX KP IS TOO SMALL TO PERMIT SOLUTION WITHIN DTC/(KP-KT
     1)UPPER LIMIT'/T10,'OPTIMIZATION NOT ATTEMPTED')
      GO TO 1260
1140  KCCO(1)=0
      KECO(1)=0
      JO=1
      CALL VMM (XI,6,XMIN,PRNT)
      CALL FUNCT (F,GI,XMIN,1,0,.TRUE.)
      WRITE (6,1150) TYP(2),DIR(1),KO,FMIN
1150  FORMAT (' PARAMETER VALUES AND DERIVATIVES FOR',5X,A8,' SEAL',3X,A
     14,' DIRECTION',I6,' KNIVES',5X,'MIN FLOW',G14.6)
      WRITE (6,1020)
      WRITE (6,1030) (B(I),XVMIN(I),XMIN(I),XVMAX(I),GI(I),I=1,6)
      T1=XMIN(1)/CL(1)
      T2=XMIN(2)*XMIN(3)/CL(1)
      T3=ADTC(1)/(XMIN(2)-XMIN(1))
      T4=ADTC(1)/CL(1)
      WRITE (6,1040)
      WRITE (6,1160) B(9),XVMIN(9),T1,XVMAX(9),B(10),XVMIN(10),T2,XVMAX
     1(10),B(11),XVMIN(11),T3,XVMAX(11),B(12),XVMIN(12),T4,XVMAX(12)
1160  FORMAT (T14,A1,T16,'KT/CL',T26,3G14.6/T14,A1,T16,'KH/CL',T26,3G14.
     16/T14,A1,T16,'DTC/(KP-KT)',T26,3G14.6/T14,A1,T16,'DTC/CL',T26,3G14
     2.6)
      T1=(XMIN(2)-XMIN(1))/(XMIN(2)*XMIN(3))
      T2=XMIN(4)/CL(1)
      WRITE (6,1170) B(13),XVMIN(13),T1,XVMAX(13),B(14),XVMIN(14),T2
     1,XVMAX(14)
1170  FORMAT (T14,A1,T16,'(KP-KT)/KH',T26,3G14.6/T14,A1,T16,'SH/CL',T26,
     13G14.6)
      WRITE (6,1060)
      WOPT(IO,JO,KO)=FMIN
      OKT(IO,JO,KO)=XMIN(1)
      OKP(IO,JO,KO)=XMIN(2)
      OKH(IO,JO,KO)=XMIN(2)*XMIN(3)
      OSH(IO,JO,KO)=XMIN(4)
      OKTH(IO,JO,KO)=XMIN(5)
      ORUF(IO,JO,KO)=XMIN(6)
      IF (FMIN.GE.WMIN) GO TO 1180
      WMIN=FMIN
      IOP=IO
      JOP=JO
      KOP=KO
1180  DO 1190 I=1,6
      XI(I)=XMIN(I)
1190  CONTINUE
1200  IF (ODIR.NE.ALTSD.AND.ODIR.NE.BOTH) GO TO 1260
C                                                  LTSD DIRECTION
      DO 1210 I=1,6
      XVMIN(I+8)=STEPMN(I,2)
```

```
      XVMAX(I+8)=STEPMX(I,2)
 1210 CONTINUE
      WRITE (6,1220) KO
 1220 FORMAT ('OPTIMIZATION STEP FOR STEPPED SEAL, LTSD DIRECTION, WITH
     1 NO. KNIVES =',I3)
      DUM=KO-1
      TKH=XI(2)*XI(3)
      IF (TKH.GE.HMAX) XI(2)=.5*HMAX/XI(3)
      IF (XLMAX.GT.0..AND.KO.GT.1) XVMAX(2)=(XLMAX-XI(1))/DUM
      IF (XI(2).GE.XVMAX(2)) XI(2)=.98*XVMAX(2)
      IF (HMAX.GT.0..AND.KO.GT.1) XVMAX(4)=(HMAX-XI(2)*XI(3))/DUM
      IF (XI(4).GE.XVMAX(4)) XI(4)=.98*XVMAX(4)
      WRITE (6,990) (I,XVMIN(I),XI(I),XVMAX(I),I=1,6)
      WRITE (6,1000) (I,XVMIN(I),XVMAX(I),I=9,14)
      TRY=ADTC(1)/(XVMAX(2)-XVMIN(1))
      IF (TRY.LE.XVMAX(11)) GO TO 1230
      WRITE (6,1130)
      GO TO 1260
 1230 KCCO(1)=0
      KECO(1)=0
      JO=2
      CALL VMM (XI,6,XMIN,PRNT)
      CALL FUNCT (F,GI,XMIN,1,0,.TRUE.)
      WRITE (6,1150) TYP(2),DIR(2),KO,FMIN
      WRITE (6,1020)
      WRITE (6,1030) (B(I),XVMIN(I),XMIN(I),XVMAX(I),GI(I),I=1,6)
      T1=XMIN(1)/CL(1)
      T2=XMIN(2)*XMIN(3)/CL(1)
      T3=ADTC(1)/(XMIN(2)-XMIN(1))
      T4=ADTC(1)/CL(1)
      WRITE (6,1040)
      WRITE (6,1160) B(9),XVMIN(9),T1,XVMAX(9),B(10),XVMIN(10),T2,XVMAX
     1(10),B(11),XVMIN(11),T3,XVMAX(11),B(12),XVMIN(12),T4,XVMAX(12)
      T1=(XMIN(2)-XMIN(1))/(XMIN(2)*XMIN(3))
      T2=XMIN(4)/CL(1)
      WRITE (6,1170) B(13),XVMIN(13),T1,XVMAX(13),B(14),XVMIN(14),T2
     1,XVMAX(14)
      WRITE (6,1060)
      WOPT(IO,JO,KO)=FMIN
      OKT(IO,JO,KO)=XMIN(1)
      OKP(IO,JO,KO)=XMIN(2)
      OKH(IO,JO,KO)=XMIN(2)*XMIN(3)
      OSH(IO,JO,KO)=XMIN(4)
      OKTH(IO,JO,KO)=XMIN(5)
      ORUF(IO,JO,KO)=XMIN(6)
      IF (FMIN.GE.WMIN) GO TO 1240
      WMIN=FMIN
      IOP=IO
      JOP=JO
      KOP=KO
 1240 DO 1250 I=1,6
      XI(I)=XMIN(I)
 1250 CONTINUE
 1260 KO=KO+1
      IF (KO.LE.KN2) GO TO 1100
 1270 CONTINUE
      WRITE (6,1280) TITLE,TITLEC
 1280 FORMAT ('
```

```
',/,4X,20A4,/,4X,20A4)
      WRITE (6,1290) CL(1),PKR(1),ADTC(1),AKBETA(1),PRATIO,PT(1),TR(1)
1290  FORMAT (' ',T40,'CLEARANCE',T55,G14.6/T40,'KNIFE RADIUS',T55,G14.6
     1/T40,'DIST TO CONTACT',T55,G14.6/T40,'BETA',T55,G14.6/T40,'PRESSUR
     2E RATIO',T55,G14.6/T40,'PRESSURE IN',T55,G14.6/T40,'TEMPERATURE',T
     355,G14.6)
      IF (XLMAX.GT.0.) WRITE (6,1300) XLMAX
1300  FORMAT (T40,'MAX SEAL LENGTH',T55,G14.6)
      IF (HMAX.GT.0.) WRITE (6,1310) HMAX
1310  FORMAT (T40,'MAX SEAL HEIGHT',T55,G14.6)
      WRITE (6,1320)
1320  FORMAT (' ',T10,'SUMMARY OF MINIMUM FLOW FOR VARIOUS SEAL CONFIGUR
     1ATIONS'/' ',T20,'NO.',T86,'SEAL',T95,'SEAL'/T4,'TYPE',T13,'DIR',T1
     28,'KNIVES',T30,'KT',T41,'KP',T51,'KH',T62,'SH',T69,'KTHETA',T77,'R
     3OUGH',T85,'LENGTH',T94,'HEIGHT',T103,'MIN FLOW')
      DO 1390 I=1,2
      DO 1380 J=1,2
      JP=J
      IF (I.EQ.1) JP=3
      DO 1370 K=KN1,KN2
      IF (K.EQ.KN1) WRITE (6,1330)
1330  FORMAT (1X)
      IF (WOPT(I,J,K).LE.0.) GO TO 1370
      DUM=K-1
      ACTL=DUM*OKP(I,J,K)+OKT(I,J,K)
      ACTH=DUM*OSH(I,J,K)+OKH(I,J,K)
      IF (I.EQ.1) WRITE (6,1340) TYP(I),DIR(JP),K,OKT(I,J,K),OKP(I,J,K)
     1,OKH(I,J,K),OKTH(I,J,K),ORUF(I,J,K),ACTL,WOPT(I,J,K)
1340  FORMAT (T2,A8,T12,A4,I6,F13.8,2F10.6,11X,F9.4,F7.2,F9.5,9X,F11.7)
      IF (I.EQ.2) WRITE (6,1350) TYP(I),DIR(JP),K,OKT(I,J,K),OKP(I,J,K)
     1,OKH(I,J,K),OSH(I,J,K),OKTH(I,J,K),ORUF(I,J,K),ACTL,ACTH,WOPT(I,J
     2,K)
1350  FORMAT (T2,A8,T12,A4,I6,F13.8,2F10.6,F11.7,F9.4,F7.2,F9.5,F9.5,F11
     1.7)
      IF (I.EQ.IOP.AND.J.EQ.JOP.AND.K.EQ.KOP) WRITE (6,1360)
1360  FORMAT ('+',T111,'----OPTIMUM')
1370  CONTINUE
1380  CONTINUE
1390  CONTINUE
      RETURN
C
1400  WRITE (6,1410)
1410  FORMAT (' END OF DATA REACHED PREMATURELY IN SUBROUTINE OPTSUB')
      STOP
      END
C     DATA SET SFCDCCNSDF AT LEVEL 004 AS OF 08/12/83
      SUBROUTINE CNSDEF (XV,WW,GC,PRNT)
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER    AOPTD*5, XTYPE*5, AOPT*5, AOPT1*5
      CHARACTER    TYPE*5,  DIRECT*5, TMIX*5
      CHARACTER    TITLE*4, TITLEC*4
      COMMON /C12R/ WVAL(30),TT(100),AA(100)
      COMMON /C12I/ NV,NQ
      COMMON /C13R/ PARVAL,AMIN,ROGCN
      COMMON /C13RA/ TITLE(20),TITLEC(20),AOPTD
      COMMON /C13I/ MG,NOSETS,ISETS
      COMMON /C14R/ G,GP1,GP1GM1,GP1OG,GP1O2,GM1O2,XK(100)
      COMMON /C15R/ C1,GP1GM2,CMAX,GM1O2D,GOGM1
      COMMON /C124R/ A(100),FFLD(100),PARM(100),AMUL(100),ADDER(100)
     1,XKPRT(100),XKPRTP(100)
      COMMON /C124I/ KURVE(100),IDPORQ(100)
      COMMON /C1234R/ PT(100),TTSR(100)
```

```
      COMMON /C1234I/ NL1,KODE
      COMMON /C234R/ V,W,FF,PS(100),XMN(100)
      COMMON /C78R/ XCL,XKP,XKH,XKT,XSH,XDTC,XDIRCT,XKR,XKTH
     1,XVGDIA,XELDIA,XLENGH,XKBETA
      COMMON /C78RA/ XTYPE
      COMMON /VECTR/ CL(30),AKR(30),AKT(30),AKP(30),AKH(30),ASH(30),ADTC
     1(30),AKTH(30),AKBETA(30),DIA(30),ROUGH(30),TR(30),A4FLOD(30),DELC
     2(30),DELE(30),KCCO(30),KECO(30),PKR(30)
      COMMON /LIMIT/ XVMIN(25),XVMAX(25)
      COMMON /KNIFE/ XLMAX,HMAX,KO,IO,JO
      COMMON /OPTDAT/ ALENGH,DMIN,DMAX,ALFA,PRATIO,XMW,ARFACT,NOW
     1,ICURVE,ICVTYP,IPRINT
      COMMON /OPTX  / AOPT
      COMMON /CROUGH/ ROUFLD(100),ROUFED(100),ROUFDH(100)
      COMMON /CMIX/ TMIX
C
      DIMENSION  XV(6), GC(25)
      CHARACTER  TYP(2)*8, ALTSD*5, STLD*5, STEP*5, STRGHT*5

C
      LOGICAL PRNT
      DATA PIE/3.14159265/
      DATA ALTSD/'LTSD '/,STLD/'STLD '/
      DATA STEP/'STEP '/,STRGHT/'STRAI'/
      DATA AOPT1/'UNITY'/
      DATA TYP/'STRAIGHT',' STEPPED'/
C
      BTAN(TDUM)=TAN(TDUM*RADIAN)
      BSIN(CDUM)=SIN(CDUM*RADIAN)
      BABS(ADUM)=ABS(ADUM)
      BSQRT(SDUM)=SQRT(SDUM)
      BMAX1(ARG1,ARG2)=DMAX1(ARG1,ARG2)
      BMIN1(ARG1,ARG2)=DMIN1(ARG1,ARG2)
C
      RADIAN=PIE/180.
      IF (IO.EQ.1) TYPE=STRGHT
      IF (IO.EQ.2) TYPE=STEP
      IF (JO.EQ.1) DIRECT=STLD
      IF (JO.EQ.2) DIRECT=ALTSD
      ASH(1)=XV(4)
      NOKNIF=KO
      AVGDIA=BMAX1(DMIN,DMAX)
      DIA(1)=AVGDIA
      KCCO(1)=-1
      IF (TYPE.EQ.STRGHT) KCCO(1)=1
      KECO(1)=-1
      IF (TYPE.EQ.STRGHT) KECO(1)=1
      IF (NOKNIF.LE.1) GO TO 50
      DUM=NOKNIF
      IF (TYPE.NE.STEP) GO TO 20
      DEL=BABS(ASH(1)*2.)
      IF (DIRECT.EQ.ALTSD) DEL=-DEL
      IF (DIRECT.EQ.ALTSD) GO TO 10
C                                                     STLD DIRECTION
      IF (DMIN.GT.0.) DIA(1)=DMIN
      IF (DMAX.GT.0.) DIA(1)=DMAX-(DUM-1.)*DEL
      GO TO 20
C                                                     LTSD DIRECTION
10    IF (DMAX.GT.0.) DIA(1)=DMAX
      IF (DMIN.GT.0.) DIA(1)=DMIN-(DUM-1.)*DEL
20    SUM=0.
      DO 40 I=1,NOKNIF
      AKT(I)=XV(1)
```

```
      AKP(I)=XV(2)
      AKH(I)=XV(2)*XV(3)
      ASH(I)=XV(4)
      AKTH(I)=XV(5)
      ROUGH(I)=XV(6)
      IF (I.LT.2) GO TO 30
      KCCO(I)=KCCO(1)
      KECO(I)=KECO(1)
      IF (TYPE.NE.STEP) DIA(I)=DIA(I-1)
      IF (TYPE.EQ.STEP) DIA(I)=DIA(I-1)+DEL
30    SUM=SUM+DIA(I)
40    CONTINUE
      AVGDIA=SUM/DUM
50    KCCO(1)=-1
      KECO(NOKNIF)=-1
C
C             CALCULATE ALPHA FOR STRAIGHT SEALS OR AMUL FOR STEP SEALS
C
      XCL=CL(1)
      XKT=AKT(1)
      XKP=AKP(1)
      XKH=AKH(1)
      XTYPE=TYPE
      ALPHA=ALFA
      XMUL=1.
C
      IF (TYPE.EQ.STRGHT) GO TO 60
      XDTC=ADTC(1)
      XSH=ASH(1)
      XKL=XKP-XKT
      RDTC=XDTC/XCL
      RKT=XKT/XCL
      RKH=XKH/XCL
      RSH=XSH/XCL
      RDTCKL=XDTC/XKL
      RKLKH=XKL/XKH
      IF (XSH.GT.XCL) XMUL=1.168*RDTC/BSQRT(RDTC**2+0.500)*RKT**0.156
     1*RDTCKL**0.109/RKH**0.0075*RKLKH**0.095/RSH**0.025
      IF (DIRECT.EQ.STLD) GO TO 70
      DUM=1.0
      IF (XSH.GT.XCL) DUM=0.89*RKH**0.061
      XMUL=XMUL*DUM
      GO TO 70
60    IF (ALPHA.GT.0.) GO TO 70
      ALPHA=0.
      XX=(XKP-XKT)/XKH
      IF (NOKNIF.GT.1) ALPHA=3.79*BSQRT(XX)
70    CONTINUE
C
      TANAL=BTAN(ALPHA)
      IF (ALPHA.GT.0.0) OOTANA=1./TANAL
C
C        SET UP RESTRICTION DATA FOR EACH KNIFE
C
      NL=-1
      DO 120 I=1,NOKNIF
      IP1=I+1
      IM1=I-1
      NL=NL+3
      NLP1=NL+1
      NLP2=NL+2
      THETA=AKTH(I)
      BETA=AKBETA(I)/2.
```

```
      BETAC=90.-BETA
      TANBET=BTAN(BETA)
      IF (ALENGH.GT.0.0) AREA=CL(I)*ALENGH
      IF (ALENGH.LE.0.0) AREA=CL(I)*PIE*(DIA(I)+CL(I))
      TT(NL)=TR(I)
      TT(NLP1)=TR(I)
      TT(NLP2)=TR(I)
      TTSR(NL)=BSQRT(TR(I))
      TTSR(NLP1)=TTSR(NL)
      TTSR(NLP2)=TTSR(NL)
      A(NL)=AREA
      A(NLP1)=AREA
      A(NLP2)=AREA
      XK(NL)=0.0
      XK(NLP1)=0.0
      XK(NLP2)=1.0
      XKPRT(NL)=0.0
      XKPRT(NLP1)=0.0
      XKPRT(NLP2)=1.0
      XKPRTP(NL)=0.0
      XKPRTP(NLP1)=0.0
      XKPRTP(NLP2)=1.0
      KURVE(NL)=2
      KURVE(NLP1)=1
      KURVE(NLP2)=0
      IDPORQ(NL)=1
      IDPORQ(NLP1)=1
      IDPORQ(NLP2)=0
      PKR(I)=ARFACT*AKR(I)
      PARM(NL)=ARFACT*AKR(I)/CL(I)
      PARM(NLP1)=AKT(I)/CL(I)
      PARM(NLP2)=0.0
      ADDER(NL)=0.0
      ADDER(NLP1)=0.0
      ADDER(NLP2)=0.0
      IF (THETA.LT.90.) ADDER(NL)=2.73*(1.-BSIN(THETA))
      DELTAC=0.0
      DELTAE=0.0
      IF (ALPHA.LE.0.0) GO TO 90
      IF (I.LE.1.OR.KCCO(I).LT.0) GO TO 80
      IF (THETA.GT.BETAC) DELTAC=(AKP(IM1)-0.5*(AKT(I)+AKT(IM1)))/
     1(OOTANA+TANBET)
      IF (THETA.LE.BETAC) DELTAC=(AKP(IM1)-0.5*(AKT(I)+AKT(IM1)))*TANAL
      IF (DELTAC.GT.AKH(I)) DELTAC=AKH(I)
80    IF (I.GE.NOKNIF.OR.KECO(I).LT.0) GO TO 90
      THETAP=AKTH(IP1)
      BETAP=AKBETA(IP1)
      BETAPC=90.-BETAP
      TANBTP=BTAN(BETAP)
      IF (THETAP.GT.BETAPC) DELTAE=(AKP(I)-0.5*(AKT(I)+AKT(IP1)))/
     1(OOTANA+TANBTP)
      IF (THETAP.LE.BETAPC) DELTAE=(AKP(I)-0.5*(AKT(I)+AKT(IP1)))*TANAL
      IF (DELTAE.GT.AKH(I+1)) DELTAC=AKH(I+1)
90    AMULTC=1.0
      AMULTE=1.0
      IF (DELTAC.GT.0.0) AMULTC=1.-CL(I)/(CL(IM1)+DELTAC)
      IF (DELTAE.GT.0.0) AMULTE=1.-CL(I)/(CL(I)+DELTAE)
      IF (AMULTC.LT.0.) AMULTC=.000001
      IF (AMULTE.LT.0.) AMULTE=.000001
      AMUL(NL)=AMULTC
      FETH=THETA
      IF (THETA.EQ.90.) FETH=THETA+BETA
      IF (FETH.GT.90.) AMUL(NL)=AMUL(NL)*(1.-.014*(FETH-90.))
```

```
      AMUL(NLP1)=BSQRT(AMULTC)*AMULTE
      AMUL(NLP2)=AMULTE**2
      DELC(I)=DELTAC
      DELE(I)=DELTAE
C
C          SET UP ROUGHNESS ARRAY VARIABLES
C
      ROUFLD(NL)=0.0
      ROUFLD(NLP1)=0.0
      ROUFLD(NLP2)=0.0
      ROUFED(NL)=0.0
      ROUFED(NLP1)=0.0
      ROUFED(NLP2)=0.0
      ROUFDH(NL)=0.0
      ROUFDH(NLP1)=0.0
      ROUFDH(NLP2)=0.0
      IF (ROUGH(I).LE.30.) GO TO 110
      ROUFDH(NLP1)=2.0*CL(I)
C               FOR STEP    SEAL LENGTH PARM. = KT
C               FOR STAIGHT SEAL LENGTH PARM. = KT FOR LAST KNIFE
      DUM=AKT(I)
      IF (TYPE.EQ.STEP.OR.I.EQ.NOKNIF) GO TO 100
      IF (TYPE.EQ.TMIX.AND.ASH(IP1).NE.0.0) GO TO 100
C               FOR STAIGHT SEAL LENGTH PARM. = KP FOR ALL BUT LAST KNIFE
      DUM=AKP(IP1)
100   ROUFLD(NLP1)=DUM/ROUFDH(NLP1)
      ROUFED(NLP1)=1.D-6*(ROUGH(I)-30.)/ROUFDH(NLP1)
110   CONTINUE
C
C            END SETTING UP ROUGHNESS ARRAY VARIABLES
C
      FFLD(NL)=0.0
      FFLD(NLP1)=0.0
      FFLD(NLP2)=0.0
      IF (A4FLOD(I).LE.0.0) GO TO 120
      FFLD(NLP1)=A4FLOD(I)
      AMUL(NLP1)=0.0
      ADDER(NLP1)=0.0
      PARM(NLP1)=0.0
      KURVE(NLP1)=0
120   CONTINUE
      NL1=NL+2
      TT(1)=TR(1)
      TTSR(1)=BSQRT(TR(1))
      A(1)=A(2)
C
C        DETERMINE MINIMUM AREA TO REFERENCE PHI IF AOPT.NE.AOPT1(UNITY)
C              'AMIN' IS REFERENCE AREA  USED IN PHI PRINTED AND PUNCHED
C
      AMIN=1.0
      IF (AOPT.EQ.AOPT1) GO TO 170
      IF (ALENGH.LE.0) GO TO 140
      AMIN=A(1)
      DO 130 I=2,NL1
      IF (A(I).LT.AMIN) AMIN=A(I)
130   CONTINUE
      GO TO 170
140   CLAVG=CL(1)
      IF (NOKNIF.LE.1) GO TO 160
      DO 150 I=2,NOKNIF
150   CLAVG=CLAVG+CL(I)
      CLAVG=CLAVG/NOKNIF
160   AMIN=CLAVG*AVGDIA*PIE
```

```
C
170   CONTINUE
C
C              MODIFY AREAS FOR XMUL    VALUE AND CL INCREASE DUE TO
C                        ROUGHNESS
C
C                   AA'S ARE ACTUAL AREAS AND A'S ARE MODIFIED AREAS USED
C                            IN FLOW CALCULATIONS
C
      DO 180 I=1,NL1
180   AA(I)=A(I)
      IF (NOSETS.GE.0.AND.TYPE.NE.STRGHT) PARVAL=XMUL
      DUM=1.0
      IF (ROUGH(1).GT.30.0) DUM=(CL(1)+ROUGH(1)*0.000001)/CL(1)
C           BMUL USED TO FORCE NO LIP IN FRONT OF FIRST KNIFE FOR STEP
      BMUL=1.
      A(1)=A(1)*BMUL*DUM
      J=1
      DO 190 I=2,NL1,3
      DUM=1.0
      IF (ROUGH(J).GT.30.0) DUM=(CL(J)+ROUGH(J)*0.000001)/CL(J)
      A(I)=A(I)*BMUL*DUM
      A(I+1)=A(I+1)*BMUL*DUM
      A(I+2)=A(I+2)*BMUL*DUM
      J=J+1
      BMUL=XMUL
190   CONTINUE
C
      I=1
      IF (TYPE.EQ.STEP) I=2
      IF (.NOT.PRNT) GO TO 290
      WRITE (6,200)
200   FORMAT (' RESULTS FOR CONVERGED VARIABLE VALUES')
      WRITE (6,210) NOKNIF,TYP(I),DIRECT,ALENGH,AVGDIA,PT(1)
210   FORMAT (1H ,'NUMBER OF KNIVES                =',I10/1H ,'SEAL TYPE
     1                    =',2X,A8/1H ,'FLOW DIRECTION
     2=',6X,A4/1H ,'SEAL LENGTH (2-D SEAL)          =',F10.4,' (INCHES)'/
     31H ,'AVG. KNIFE DIAMETER (3-D SEAL) =',F10.4,' (INCHES)'/1H ,'INLE
     4T TOTAL PRESSURE            =',F10.4,' (PSIA)')
      IF (AOPT.EQ.AOPT1) WRITE (6,220) AMIN
220   FORMAT (' AREA NORMALIZING FACTOR         =',F10.4)
      WRITE (6,230)
230   FORMAT (1H ,4X, 'K N I F E   G E O M E T R Y   D A T A'/' KNIFE
     1CL     KR      KT     KP     KH     SH     DTC  ','THETA  BETA   D
     2IA   ROUGH  TEMP  KCCO KECO 4FL/D ',' DEL C  DEL E  AREA  ALPHA')
      WRITE (6,240)
240   FORMAT (1H ,'  NO.  (IN)    (IN)    (IN)    (IN)    (IN)    (IN)    (IN
     1)   ','(DEG) (DEG)  (IN)    (RMS) (DEGR)                  ','  (IN)
     2 (IN)  MULT  (DEG)')
      WRITE (6,250)
250   FORMAT (' ----- ------ ------- ------ ------ ------ ------ ------
     1','----- ----- ------  ----- ------ ---- ---- ----- ','------ ----
     2-- -----  -----'/)
      BMUL=1.
      BDTC=0.
      BSH=0.
      DO 280 I=1,NOKNIF
      IF (I.GT.1) BSH=ASH(I)
      IF (I.GT.1) BDTC=ADTC(I)
      IF (I.GT.1) BMUL=XMUL
      IF ( I .EQ. 1 )
     +WRITE (6,260) I,CL(I),PKR(I),AKT(I),AKP(I),AKH(I),BSH,BDTC,AKTH(I)
     +,AKBETA(I),DIA(I),ROUGH(I),TR(I),KCCO(I),KECO(I),A4FLOD(I),DELC(I)
```

```
     +,DELE(I),BMUL
      IF ( I .GT. 1 )
     +WRITE (6,260) I,CL(I),PKR(I),AKT(I),AKP(I),AKH(I),BSH,BDTC,AKTH(I)
     +,AKBETA(I),DIA(I),ROUGH(I),TR(I),KCCO(I),KECO(I),A4FLOD(I),DELC(I)
     +,DELE(I),BMUL, ALPHA
260   FORMAT (I5,F8.4,F8.5,5F7.4,2F6.1,F7.4,F7.2,F7.1,I4,I5,F8.2,2F7.4,
     1         2F6.3)
280   CONTINUE
C
C        CALL SUBROUTINE CHOKEC  TO FIND CHOKE POINT OR USE W'S READ IN
C
290   NQ=IPRINT
      MG=ICVTYP
      NV=NOW
      IF (NV.GT.0) NQ=1
      CALL CHOKEC (PRATIO,WW,PRNT,.TRUE.)
      IF (TYPE.EQ.STEP) GO TO 300
C                                          STRAIGHT SEAL CONSTRAINTS
      TRY=XV(1)/CL(1)
      GC(7)=BMIN1(1.-TRY/XVMAX(7),TRY/XVMIN(7)-1.)
      TRY=(XV(2)-XV(1))/(XV(3)*XV(2))
      GC(8)=BMIN1(1.-TRY/XVMAX(8),TRY/XVMIN(8)-1.)
      GO TO 310
C                                                STEPPED SEAL CONSTRAINTS
300   TRY=XV(1)/CL(1)
      GC(9)=BMIN1(1.-TRY/XVMAX(9),TRY/XVMIN(9)-1.)
      TRY=XV(2)*XV(3)/CL(1)
      GC(10)=BMIN1(1.-TRY/XVMAX(10),TRY/XVMIN(10)-1.)
      TRY=ADTC(1)/(XV(2)-XV(1))
      GC(11)=BMIN1(1.-TRY/XVMAX(11),TRY/XVMIN(11)-1.)
      TRY=ADTC(1)/CL(1)
      GC(12)=BMIN1(1.-TRY/XVMAX(12),TRY/XVMIN(12)-1.)
      TRY=(XV(2)-XV(1))/(XV(3)*XV(2))
      GC(13)=BMIN1(1.-TRY/XVMAX(13),TRY/XVMIN(13)-1.)
      TRY=XV(4)/CL(1)
      GC(14)=BMIN1(1.-TRY/XVMAX(14),TRY/XVMIN(14)-1.)
C                                                      ROUGHNESS CONSTRAINT
310   TRY=(ROUGH(1)-30.)/(2.*CL(1))
      GC(15)=BMIN1(1.-TRY/XVMAX(15),TRY/10.-1.)
      GC(15)=GC(15)/1000.
      RETURN
      END
```

Program Listing

File: K22.FOR

```
C -----------------------------------------------------------------
C     File:    K22.FOR
C     Program: KTK
C     Version: 2
C
C     Revision History:
C     (1) Modified December 1993 to compile under OS/2 using the
C     WATCOM F77/386 V9.0 compiler.
C -----------------------------------------------------------------
C     DATA SET SFCDCFUNCT AT LEVEL 004 AS OF 08/12/83
      SUBROUTINE FUNCT (F,G,X,IDER,ICON,PRNT)
      IMPLICIT REAL*8(A-H,O-Z)
C        F IS THE PAYOFF FUNCTION DEFINED BY FUNCT.
C        G IS THE VECTOR OF DERIVATIVES OF F WITH RESPECT TO X DEFINED
C          BY FUNCT.
C        X IS THE INDEPENDENT VARIABLE VECTOR OF LENGTH NV DEFINED BY
C          THE CALLING PROGRAM.
C        IDER IS A TRIGGER TO INDICATE DERIVATIVES ARE REQUIRED.
C          1 = CALCULATE DERIVATIVES.  0 = DON'T.
C        ICON IS A TRIGGER TO IGNORE CONSTRAINTS IN DERIV CALC
C          0 MEANS IGNORE,  NONZER0 MEANS USE CONSTRAINTS
      DIMENSION X(6), G(6), GCSAVE(25,6), WSAVE(6), XV(6), GC(25)
      COMMON /OPTION/ IFIX(6),IOFFC(25), IDOMAN,IDAVLM,IDAVE,ISRCHP
     1,IPASSP
      COMMON /OPTL/ ONEPT

C        IFIX IS A VECTOR OF TRIGGERS INDICATING WHETHER TO FREEZE
C          INDEPENDENT VARIABLES AT THEIR INITIAL VALUES.  1 = FREEZE.
C        IOFFC IS A VECTOR OF TRIGGERS INDICATING DISABLED CONSTRAINTS.
C        ONEPT INDICATES A NON-OPTIMIZATION RUN.  EVALUATE F ONLY.
C        IDOMAN WILL BE SET TO 1 IF ALL CONSTRAINTS ARE SATISFIED.
C        IDAVLM IS THE LIMITING PASS NUMBER (IDAVE)
      COMMON /LIMIT/ XVMIN(25),XVMAX(25)
C        XVMIN AND XVMAX ARE USER READ IN LIMITS FOR THE INDEPENDENT
C          VARIABLES.
      COMMON /KNIFE/ XLMAX,HMAX,KO,IO,JO
      COMMON /OPTMUM/ FMIN
      CHARACTER B*4
      CHARACTER BLNK*4, STAR*4
      COMMON /BIND/ B(25)
      LOGICAL ONEPT,PRNT,PRT,SETUP
      DATA SETUP/.FALSE./

      DATA TWK/.1E-6/,BLNK/'    '/,STAR/'****'/
      BMIN1(ARG1,ARG2)=DMIN1(ARG1,ARG2)
C
C     NC and NV moved here for initialization at each call
C
      NV=6
      NC=15
      IF (SETUP) GO TO 30
      SETUP=.TRUE.
C      NV=6
C      NC=15
      DO 10 I=1,NC
10    GC(I)=0.
      DO 20 I=1,NV
      IF (XVMAX(I).LE.XVMIN(I)) IFIX(I)=1
      IF (IFIX(I).EQ.1) IOFFC(I)=1
20    CONTINUE
30    IVAR=NV+1
      IF (IOFFC(1).GT.0) IOFFC(7)=1
      IF (IOFFC(1).GT.0) IOFFC(9)=1
```

```
      IF (IOFFC(1).GT.0.AND.IOFFC(2).GT.0) IOFFC(11)=1
      IF (IOFFC(2).GT.0.AND.IOFFC(3).GT.0) IOFFC(10)=1
      IF (IOFFC(1).GT.0.AND.IOFFC(10).GT.0) IOFFC(13)=1
      IF (IOFFC(1).GT.0.AND.IOFFC(10).GT.0) IOFFC(8)=1
      IF (IOFFC(4).GT.0) IOFFC(14)=1
      IF (IOFFC(6).GT.0) IOFFC(15)=1
      IF (IDER.EQ.0) GO TO 70
      IVAR=1
40    XSAVE=X(IVAR)
C        THIS BLOCK OF LOGIC CONTROLS NUMERICAL DERIVATIVE COMPUTATION.
      X(IVAR)=X(IVAR)*(1.+TWK)
      G(IVAR)=0.
      IF (X(IVAR).EQ.0.) GO TO 50
      IF (IFIX(IVAR).EQ.1) GO TO 50
      GO TO 70
50    CONTINUE
      DO 60 K=1,NC
      GCSAVE(K,IVAR)=GC(K)
60    CONTINUE
      WSAVE(IVAR)=W
      X(IVAR)=XSAVE
      IVAR=IVAR+1
      IF (IVAR.LE.NV) GO TO 40
70    CONTINUE
C        COPY THE INDEPENDENT VARIABLES INTO XV.  XV MAY BE IN COMMON.
      DO 80 I=1,NV
80    XV(I)=X(I)
C        ****************************************************************
      IF (KO.LE.1) GO TO 90
      DUM=KO-1
      IF (XLMAX.GT.0..AND.IFIX(2).LT.1) XVMAX(2)=(XLMAX-XV(1))/DUM
      IF (IO.NE.2) GO TO 90
      IF (HMAX.GT.0..AND.IFIX(4).LT.1) XVMAX(4)=(HMAX-XV(2)*XV(3))/DUM
90    CONTINUE
      PRT=.FALSE.
      IF (ICON.EQ.0.AND.IVAR.GT.NV) PRT=PRNT
      CALL CNSDEF (XV,W,GC,PRT)
      FMIN=W
C         DEFINE THE OBJECTIVE FUNCTION AND CONSTRAINTS AT XV
C        ****************************************************************
C        GC(1 - NV) XVMIN - XVMAX LIMITS ARE DEFINED HERE.
      DO 100 I=1,NV
      GC(I)=BMIN1(1.-XV(I)/XVMAX(I),XV(I)/XVMIN(I)-1.)
100   CONTINUE
      GC(2)=GC(2)/10.
      IF (IVAR.LE.NV) GO TO 50
C        EVALUATE THE PAYOFF FUNCTION
C                              SCALE F TO BE COMPATIBLE WITH PENALTIES
      F=W*10000.
      IDOMAN=1
      DO 110 I=1,NC
      IF (IOFFC(I).EQ.1) GO TO 110
      IF (GC(I).GE..001D0) GO TO 110
      F=F+(1.-1000.*GC(I))**2
      IF (GC(I).LT.0.) IDOMAN=0
110   CONTINUE
      F=.5*F
      IF (ONEPT) GO TO 140
C     IF (IDER.EQ.0) GO TO 86      TO PRINT EVERY TIME THROUGH.
      IF (IDER.EQ.0) RETURN
      DO 130 J=1,NV
      IF (X(J).EQ.0.) GO TO 130
      IF (IFIX(J).EQ.1) GO TO 130
```

```
      G(J)=.5*(WSAVE(J)-W)/(TWK*X(J))*10000.
      IF (ICON.LE.0) GO TO 130
      DO 120 I=1,NC
      IF (IOFFC(I).EQ.1) GO TO 120
      IF (GC(I).LT..001D0) G(J)=G(J)-1000.*(1.-1000.*GC(I))*(GCSAVE(I,J)
     1-GC(I))/(TWK*X(J))
120   CONTINUE
130   CONTINUE
      IF (ICON.GT.0) RETURN
140   CONTINUE
C     WRITE (6,150) IDAVE,F,(X(I),I=1,NV)
C 150 FORMAT (/' IDV,F,X',I5,F15.7,F15.7/(5F15.7))
      DO 180 J=1,NC
      B(J)=BLNK
      IF (IOFFC(J).EQ.1) GO TO 180
      IF (GC(J).GT..01) GO TO 180
C     WRITE (6,160) J,GC(J),XVMIN(J),XVMAX(J)
C 160 FORMAT (' BINDING SCALAR CONSTRAINT',I5,G18.11,5X,'MIN',G18.11,'
C    1 MAX',G18.11)
      B(J)=STAR
C     IF (J.LE.6) WRITE (6,170) XV(J)
C 170 FORMAT ('+',T100,'VAL',G18.11)
180   CONTINUE
      RETURN
      END


C     DATA SET SFCDCVMMPK AT LEVEL 005 AS OF 08/11/83
      SUBROUTINE VMM (X,NV,XMIN,PRNT)
      IMPLICIT REAL*8(A-H,O-Z)
C                                     DAVIDON VARIABLE METRIC METHOD
      DIMENSION X(6), G(6), XMIN(6), GMIN(6), S(6), H(21)
      DIMENSION SIG(6), HYMYH(6), Y(6), FSAV(10)
      LOGICAL ONEPT,PRNT
      COMMON /OPTION/ IFIX(6),IOFFC(25), IDOMAN,IDAVLM,IDAVE,ISRCHP
     1,IPASSP
      COMMON /OPTL/   ONEPT
      KMAP(I,J)=(IABS(I-J)*(I+J-3)+I*I+I+J*J+J)/4
      DATA EPS/0.000001/,EPS1/0.0005/
      DATA FSAV/0.,0.,0.,0.,0.,0.,0.,0.,0.,0./
C                               IDAVE = NUMBER OF DAVIDON PASSES
C                               NOFEV = NUMBER OF FUNCTION EVALUATIONS
      IGDS=0
      IDAVE=0
      NOFEV=0
      IDOM=0
      IRSET=1
C      THE F'S AND G'S ARE EVALUATED FOR THE BEGINNING OF A DAVIDON PASS
      CALL FUNCT (F,G,X,1,1,PRNT)
      IF (ONEPT) RETURN
      NOFEV=NOFEV+1
      IACTV=0
      DO 10 I=1,NV
      IF (G(I).NE.0.) IACTV=IACTV+1
10    CONTINUE
      NOFEV=NOFEV+IACTV
      NOFEV=0
20    DO 40 I=1,NV
      DO 30 J=1,I
      IJ=KMAP(I,J)
30    H(IJ)=0.
40    H(IJ)=1.
      IRSET=IDAVE+1
```

```
C          X(I) = THE VECTOR OF VARIABLES
C          F = THE FUNCTION BEING OPTIMIZED
C          G(I) = THE VECTOR OF FIRST PARTIAL DERIVATIVES OF F
C          THE FOLLOWING SEQUENCE COMPUTES THE DAVIDON STEP
C          ALONG WITH XNORM, THE NORM OF THE VARIABLES AT THE BASE POINT
C          AND SNORM, THE NORM OF THE STEP AT THE BASE POINT
50      XNORM=0.0
        SNORM=0.0
        DO 70 I=1,NV
        S(I)=0.0
        DO 60 J=1,NV
        IJ=KMAP(I,J)
60      S(I)=S(I)-H(IJ)*G(J)
        XNORM=XNORM+X(I)**2
70      SNORM=SNORM+S(I)**2
        IF (SNORM.EQ.0.) RETURN
        XNORM=SQRT(XNORM)
        SNORM=SQRT(SNORM)
        IDAVE=IDAVE+1
C
C            AFTER COMPUTATION OF THE DAVIDON DIRECTION VECTOR,S(I),IT IS
C            NECESSARY TO SEARCH ALONG THIS VECTOR TO FIND THE MINIMUM
C
        CALL SEARCH (NV,F,G,X,S,SNORM,FMIN,GMIN,XMIN,ETA,GDS,GMDS,NOFEV
       1,IBAD,NSTEP,IDAVE,ISRCHP,PRNT)
        IF (IBAD.EQ.1) GO TO 290
        IF (FMIN.EQ.0.) RETURN
        NOFEV=NOFEV+IACTV
C
C          WHEN THE MINIMUM HAS BEEN DETERMINED VIA THE LINEAR SEARCH,-THE
C          DERIVATIVES AT THIS POINT,I.E. THE GMIN(I)'S, ARE USED TO MODIFY
C          THE H-MATRIX
C
        IF (NLOOP.EQ.1.AND.IDOMAN.EQ.1) RETURN
        IF (-ABS(GMDS).GT.0.1*GDS) GO TO 80
        IRSET=IRSET+1
        GO TO 130
80      DO 90 I=1,NV
        Y(I)=GMIN(I)-G(I)
90      SIG(I)=ETA*S(I)
        SIGDY=0.
        YMHMY=0.
        DO 110 I=1,NV
        HYMYH(I)=0.
        SIGDY=SIGDY+SIG(I)*Y(I)
        DO 100 J=1,NV
        IJ=KMAP(I,J)
100     HYMYH(I)=HYMYH(I)+H(IJ)*Y(J)
110     YMHMY=YMHMY+HYMYH(I)*Y(I)
        DO 120 I=1,NV
        DO 120 J=I,NV
        IJ=KMAP(I,J)
120     H(IJ)=H(IJ)+SIG(I)*SIG(J)/SIGDY-HYMYH(I)*HYMYH(J)/YMHMY
130     DO 140 I=1,NV
        G(I)=GMIN(I)
140     X(I)=XMIN(I)
        IF (IPASSP.EQ.0) GO TO 190
C       WRITE (1,201) FMIN,ETA,GDS,GMDS
C       201 FORMAT ('FMIN,ETA,GDS,GMDS',4G15.7)
        WRITE (6,150) F,FMIN,ETA
150     FORMAT (1H ,3X,3HF =,D23.15,8H, FMIN =,D23.15,7H, ETA =,E15.7)
        WRITE (6,160) NSTEP
160     FORMAT (/3X,8HNSTEP = ,I2)
```

```
      WRITE (6,170) IDAVE,NOFEV,XNORM,SNORM,GDS,GMDS
170   FORMAT ('   IDAVE',I6,'  NOFEV',I6,'  XNORM',E15.5,'  SNORM',E15.5
     1,'  GDS',E15.5,'  GMDS',E15.5)
      WRITE (6,180) (XMIN(I),I=1,NV)
180   FORMAT (1H ,10X,21HINDEPENDENT VARIABLES,/,(10X,5E18.11))
190   CONTINUE
      IF (FMIN.LE.F) GO TO 210
      DO 200 I=1,9
200   FSAV(I)=FSAV(I)+(FMIN-F)
      F=FMIN
210   DO 220 J=1,9
      I=10-J
220   FSAV(I+1)=FSAV(I)
      FSAV(1)=F
      F=FMIN
      IF (IDAVE.GE.IDAVLM) RETURN
      DO 230 I=1,NV
      IF (ABS(G(I)*X(I)).GT.FMIN*EPS1) GO TO 280
      IF (ABS(G(I)*X(I)).GT.FMIN*1.E-04) GO TO 280
230   CONTINUE
240   WRITE (6,250)
250   FORMAT (///'  N O R M A L   C O N V E R G E N C E'//)
      RETURN
260   WRITE (6,270)
270   FORMAT (' STOPPED FOR INSUFFICIENT CUMULATIVE IMPROVEMENT.')
      RETURN
280   IF (IRSET.GT.IDAVE) GO TO 20
      IF (GMDS.EQ.GDS) GO TO 20
      IF (GMDS.LT.GDS) GO TO 50
      IF (IDAVE-IRSET.LT.9) GO TO 50
      IF (FSAV(10).LT.(1.+EPS)*F) GO TO 260
      IF ((-1.0D0)*GDS*DMAX1(ETA,1.0D0) .LT. FMIN*EPS) GO TO 240
      GO TO 50
290   IGDS=IGDS+1
      IF (IGDS.LT.5) GO TO 310
      WRITE (6,300)
300   FORMAT (' IGDS= 5')
      ONEPT=.TRUE.
      CALL FUNCT (F,G,XMIN,0,1,.FALSE.)
C     CALL PRTOUT
      CALL FUNCT (F,G,X,0,1,.FALSE.)
C     CALL PRTOUT
C     CALL EXIT
      STOP 'EXIT'
      RETURN
310   CALL FUNCT (F,G,X,1,1,.FALSE.)
      GO TO 20
      END
      SUBROUTINE SEARCH (NV,F,G,X,S,SNORM,FMIN,GMIN,XMIN,ETA,GDS,GMDS
     1,NOFEV,IBAD,NSTEP,IDAVE,IPRT2,PRNT)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL PRNT,PRT
      DIMENSION G(1), X(1), S(1), GMIN(1), XMIN(1), FV(6), XV(6), DER(2)
     1, DDY(18), TX(6), TY(6)
      DATA BOUND /0.0/
      IBAD=0
      PRT=PRNT
      IF (IDAVE.GT.1) PRT=.FALSE.
C
C                              COMPUTE  G DOT S
C
      GDS=0.
      DO 10 I=1,NV
```

```
10    GDS=GDS+S(I)*G(I)
C
C                          PRINT START POINT DATA ON PRINT TRIGGER
C
      IF (IPRT2.NE.1) GO TO 60
      WRITE (6,20) IDAVE,F,GDS,SNORM
20    FORMAT(1H1,6X,'IDAVE=',I4,' F=',E18.10,' GDS=',E15.7,' SNORM=',E13
     1.5//)
      WRITE (6,30) (X(I),I=1,NV)
30    FORMAT (6X,'INDEPENDENT VARIABLES ARE',/,(5E20.11))
      WRITE (6,40) (G(I),I=1,NV)
40    FORMAT (//,6X,'PARTIAL DERIVATIVES ARE',/,(5E20.11))
      WRITE (6,50) (S(I),I=1,NV)
50    FORMAT (//,6X,'DIRECTION VECTOR COMPONENTS ARE',/,(5E20.11))
60    CONTINUE
C
C                          IF G DOT S IS POSITIVE SOMETHING IS WRONG
C
      IF (GDS.LT.0.) GO TO 80
      WRITE (6,70)
70    FORMAT (' GDS IS POSITIVE')
      IBAD=1
      RETURN
C
C                                  INITIALIZE COUNTERS
C
80    IBOM=0
      ICUT=0
      IBF=0
      INTP=0
      NPT1=0
C
C                                   COMPUTE STEP SIZE
C
      RELMAX=0.01
      DO 90 I=1,NV
90    IF (X(I).NE.0.) RELMAX=DMAX1(RELMAX,ABS(S(I)/X(I)))
      STEP=.04/RELMAX
      IF (IDAVE.GT.1.AND.BOUND.LT.F) STEP=SQRT(0.5*STEP*(BOUND-F)/GDS)
      XV(1)=0.
      FV(1)=F
C
C                                  TAKE A STEP IN THE S DIRECTION
C
      ETA=STEP
      DO 100 I=1,NV
100   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,0,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      NPT=2
      NSTEP=1
      NOFEV=NOFEV+1
      XV(2)=ETA
      FV(2)=FMIN
      IF (IPRT2.NE.1) GO TO 120
      WRITE (6,110) NSTEP,NOFEV,IBOM,ICUT,IBF,INTP,NPT,ETA,FMIN
110   FORMAT(//,' NSTEP=',I3,' NOFEV=',I4,' IBOM=',I2,' ICUT=',I2,' IBF=
     1',I2,' INTP=',I3,' NPT=',I1,'  ETA=',E19.11,' F=',E20.11)
      WRITE (6,30) (XMIN(I),I=1,NV)
120   CONTINUE
C
C                                  HAVE WE BOXED IN A MIN
C
```

```
      IF (F-FMIN) 220,220,130
C
C                              NO MIN.  ACCELERATE STEP AND TRY AGAIN
C
130   STEP=2.*STEP
140   ETA=ETA+STEP
      DO 150 I=1,NV
150   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,0,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      NPT=NPT+1
      XV(NPT)=ETA
      FV(NPT)=FMIN
      NOFEV=NOFEV+1
160   NSTEP=NSTEP+1
      IF (IPRT2.NE.1) GO TO 170
      WRITE (6,110) NSTEP,NOFEV,IBOM,ICUT,IBF,INTP,NPT,ETA,FMIN
      WRITE (6,30) (XMIN(I),I=1,NV)
170   CONTINUE
C
C                                             HAVE WE BOXED IN A MIN
C
      IF (FV(NPT-1)-FV(NPT)) 270,180,180
180   IF (NPT-6) 210,190,190
190   DO 200 I=1,5
      XV(I)=XV(I+1)
200   FV(I)=FV(I+1)
      NPT=5
210   NBEST=NPT
      IF (NSTEP.GT.19) GO TO 790
      IF (IBOM.GT.0.AND.NPT.GE.4) GO TO 290
      IF (IBOM) 130,130,140
220   ETA=-0.5*GDS*XV(2)**2/(FV(2)-FV(1)-XV(2)*GDS)
      ETA=DMAX1(ETA,0.25*STEP)
      XV(3)=XV(2)
      FV(3)=FV(2)
      XV(2)=ETA
      DO 230 I=1,NV
230   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,0,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      FV(2)=FMIN
      NOFEV=NOFEV+1
      INTP=INTP+1
      NPT=3
      IF (IPRT2.NE.1) GO TO 250
      WRITE (6,240)
240   FORMAT (//,' QUADRATIC FIT ON FIRST STEP')
      WRITE (6,110) NSTEP,NOFEV,IBOM,ICUT,IBF,INTP,NPT,ETA,FMIN
      WRITE (6,30) (XMIN(I),I=1,NV)
250   CONTINUE
      IF (FV(2).LT.FV(1)) GO TO 270
      STEP=ETA
      ICUT=ICUT+1
      IF (ICUT.LT.10) GO TO 220
      WRITE (6,260)
260   FORMAT (' QUADRATIC INTERPOLATOR FAILURE')
      IBAD=1
      RETURN
270   IBOM=IBOM+1
      NBEST=NPT-1
C
C            MIN BOXED IN   CHECK ORDER OF MAGNITUDE OF FUNCTION VALUES
```

```
C
      IF (DLOG(FV(NPT)/FV(1))**2-5.2946) 340,280,280
280   ICUT=ICUT+1
      IF (ICUT-5) 300,300,340
290   NPT=5
      GO TO 270
C
C                              HALVE INTERVAL AND SAVE BEST POINTS
C
300   STEP=XV(NBEST)-XV(NBEST-1)
      ETA=XV(NBEST)-0.5*STEP
      STEP=0.5*(STEP+XV(NBEST+1)-XV(NBEST))
      XV(1)=XV(NBEST-1)
      FV(1)=FV(NBEST-1)
      XV(6)=XV(NBEST+1)
      FV(6)=FV(NBEST+1)
      XV(3)=XV(NBEST)
      FV(3)=FV(NBEST)
      XV(5)=XV(6)
      FV(5)=FV(6)
      DO 310 I=1,NV
310   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,0,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      XV(2)=ETA
      FV(2)=FMIN
      NPT=3
      NOFEV=NOFEV+1
      IF (IPRT2.NE.1) GO TO 330
      WRITE (6,320)
320   FORMAT (//,' ORDER OF MAGNITUDE OR BAD SPLINE FIT SEQUENCE')
330   CONTINUE
      GO TO 160
C
C             MAKE A SPLINE FIT ON UP TO SIX POINTS AND PREDICT THE MIN
C
340   DO 350 I=1,3
      TX(I)=XV(I)
      TY(I)=FV(I)
      J=NPT-I+1
      TX(I+3)=XV(J)
350   TY(I+3)=FV(J)
      K=1
360   DELX1=TX(K+1)-TX(K)
      DELX2=TX(K+2)-TX(K)
      DELX21=TX(K+2)-TX(K+1)
      DELY1=TY(K+1)-TY(K)
      DELY2=TY(K+2)-TY(K)
      A=(DELY2*DELX1-DELY1*DELX2)/(DELX1*DELX2*DELX21)
      B=DELY1/DELX1-A*(TX(K+1)+TX(K))
      IF (K.GE.4) GO TO 390
      IF (TY(1).EQ.TY(2)) GO TO 380
      DER(1)=2.*A*TX(1)+B
370   K=4
      GO TO 360
380   DER(1)=0.
      GO TO 370
390   IF (TY(5).EQ.TY(6)) GO TO 400
      DER(2)=2.*A*TX(4)+B
      GO TO 410
400   DER(2)=0.
410   CONTINUE
      DO 420 I=2,NPT
```

```
420   DDY(I)=XV(I)-XV(I-1)
      NO=NPT-1
      M=2*NPT-1
      B=DDY(2)
      C=0.5*B
      E=3.*((FV(2)-FV(1))/DDY(2)-DER(1))
      DDY(NPT+1)=C/B
      DDY(M+1)=E/B
      DO 430 I=2,NO
      K=NPT+I
      KK=M+I
      A=0.5*DDY(I)
      B=DDY(I)+DDY(I+1)
      C=0.5*DDY(I+1)
      E=3.*((FV(I+1)-FV(I))/DDY(I+1)-(FV(I)-FV(I-1))/DDY(I))
      W=B-A*DDY(K-1)
      DDY(K)=C/W
430   DDY(KK)=(E-A*DDY(KK-1))/W
      A=0.5*DDY(NPT)
      B=DDY(NPT)
      E=3.*(DER(2)-(FV(NPT)-FV(NPT-1))/DDY(NPT))
      W=B-A*DDY(M)
      DDY(NPT)=(E-A*DDY(3*NPT-2))/W
      DO 440 I=2,NPT
      K=NPT+1-I
      KK=M+K
      KKK=NPT+K
440   DDY(K)=DDY(KK)-DDY(KKK)*DDY(K+1)
      IF (NPT-3) 450,450,480
450   ETA=0.5*(XV(2)+XV(1))+(FV(1)-FV(2))/(DDY(1)*(XV(2)-XV(1)))
      IF (ETA-XV(2)) 460,460,470
460   NPT1=1
      GO TO 540
470   NPT1=2
      GO TO 540
480   KEY=0
      NPTM1=NPT-1
      DO 520 MPT1=1,NPTM1
      MPT11=MPT1+1
      A=XV(MPT1)*DDY(MPT11)-XV(MPT11)*DDY(MPT1)
      C1P=FV(MPT11)-FV(MPT1)-(DDY(MPT11)-DDY(MPT1))*(XV(MPT11)**2+XV
     1(MPT11)*XV(MPT1)+XV(MPT1)**2)/6.+.5*A*(XV(MPT1)+XV(MPT11))
      DET=A*A-2.*(DDY(MPT11)-DDY(MPT1))*C1P
      IF (DET) 520,490,490
490   XMINM=(A+SQRT(DET))/(DDY(MPT11)-DDY(MPT1))
      IF (XMINM-XV(MPT1)) 520,500,500
500   IF (XMINM-XV(MPT11)) 510,510,520
510   KEY=KEY+1
      NPT1=MPT1
      ETA=XMINM
520   CONTINUE
      IF (KEY-1) 530,540,530
530   NPT1=6
540   CONTINUE
      IF (NPT1.GT.NBEST) NPT1=6
      IF (NPT1.LT.NBEST-1) NPT1=6
      IF (IPRT2.NE.1) GO TO 560
      WRITE (6,550) XV,FV,ETA,NPT,NPT1,NBEST
550   FORMAT(//,'  XV =',6E18.10,/,'  FV =',6E18.10,//,10X,'ETA=',
     1       E18.11,'NPT=',I2,' NPT1=',I2,' NBEST=',I2)
560   CONTINUE
C
C         IF SPLINE FIT BAD CHECK FOR LOOSE CONVERGENCE AND REINITIALIZE
```

```
C
      IF (NPT1-NPT) 600,570,570
570   IBF=IBF+1
      IF (IBF-6) 580,590,590
580   IF ((XV(NBEST+1)-XV(NBEST-1))/XV(NBEST)-0.1E-2) 790,790,300
590   NSTEP=25
      GO TO 790
C
C             IF PREDICTED POINT IS NEAR THE END OF AN INTERVAL
C             STEP ACROSS THE MIN AND GET A BETTER POINT DISTRIBUTION
C
600   IF (NPT1-NPT+5) 610,610,620
610   ETA=0.5*(XV(1)+XV(2))
      NPT1=1
      GO TO 650
620   IF (NPT1-5) 640,630,630
630   ETA=0.5*(XV(5)+XV(6))
      NPT1=5
      GO TO 650
C
C                          APPLY CONVERGENCE TEST TO PREDICTED ETA
C
640   IF (ABS(XV(NBEST)-ETA).LT.1.E-3*ETA) GO TO 790
      W=0.1*(XV(NPT1+1)-XV(NPT1))
      IF (ETA-XV(NPT1).LT.W) ETA=XV(NPT1)+W
      IF (XV(NPT1+1)-ETA.LT.W) ETA=XV(NPT1+1)-W
650   DO 660 I=1,NV
660   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,0,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      NOFEV=NOFEV+1
      IF (IPRT2.NE.1) GO TO 670
      WRITE (6,110) NSTEP,NOFEV,IBOM,ICUT,IBF,INTP,NPT,ETA,FMIN
      WRITE (6,30) (XMIN(I),I=1,NV)
670   CONTINUE
C
C          INSERT ETA AND F (ETA) INTO THE SPLINE INTERPOLATION TABLE
C
      IF (NPT-6) 680,700,700
680   NPT2=NPT-NPT1
      DO 690 I=1,NPT2
      NPTI=NPT-I
      XV(NPTI+2)=XV(NPTI+1)
690   FV(NPTI+2)=FV(NPTI+1)
      XV(NPT1+1)=ETA
      FV(NPT1+1)=FMIN
      NPT=NPT+1
      GO TO 760
700   NBAD=1
      IF (FV(1).LT.FV(6)) NBAD=6
      IF ((NPT1-3)/2.NE.0) NBAD=6-NPT1
      IF (NPT1-3) 720,730,710
710   IF (FMIN.LT.FV(5)) GO TO 730
      GO TO 740
720   IF (FMIN.LT.FV(2)) GO TO 730
      NBAD=6
730   IF (NBAD.LT.2) GO TO 740
      NPT=5
      GO TO 680
740   NPT2=NPT1-1
      DO 750 NPTI=1,NPT2
      XV(NPTI)=XV(NPTI+1)
750   FV(NPTI)=FV(NPTI+1)
```

```
      XV(NPT1)=ETA
      FV(NPT1)=FMIN
760   FMIN=FV(1)
      NBEST=1
      DO 780 I=2,NPT
      IF (FMIN-FV(I)) 780,770,770
770   FMIN=FV(I)
      NBEST=I
780   CONTINUE
      INTP=INTP+1
      IF (INTP-20) 840,790,790
C
C        COMPUTE THE FUNCTION WITH DERIVATIVES AT THE MIN FOR USE BY VMM
C
790   ETA=XV(NBEST)
      DO 800 I=1,NV
800   XMIN(I)=X(I)+ETA*S(I)
      CALL FUNCT (FMIN,GMIN,XMIN,1,1,PRT)
      IF (FMIN.EQ.0.) RETURN
      NOFEV=NOFEV+1
      GMDS=0.
      DO 810 I=1,NV
810   GMDS=GMDS+GMIN(I)*S(I)
      IF (IPRT2.NE.1) GO TO 830
      WRITE (6,820)
820   FORMAT (//,' THE FOLLOWING DATA IS RETURNED BY SEARCH')
      WRITE (6,110) NSTEP,NOFEV,IBOM,ICUT,IBF,INTP,NPT,ETA,FMIN
      WRITE (6,30) (XMIN(I),I=1,NV)
      WRITE (6,40) (GMIN(I),I=1,NV)
830   CONTINUE
      BOUND=FMIN**2/F
      RETURN
840   IF (NBEST-NPT+4) 610,610,850
850   IF (NBEST.LT.5) GO TO 340
      GO TO 630
      END
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 2005	Final Contractor Report

4. TITLE AND SUBTITLE

Source Listings for Computer Code KTK
Labyrinth Seal Design Model

5. FUNDING NUMBERS

WU–506–42–31–00
WU–590–21–11–00
NAS3–25644

6. AUTHOR(S)

Mechanical Technology, Inc.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Mechanical Technology, Inc.
968 Albany Shaker Road
Latham, New York 12110

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13626

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2003-212366
MTI 94TM7

11. SUPPLEMENTARY NOTES

Project Manager, Anita D. Liang, Aeropropulsion Projects Office, NASA Glenn Research Center, organization code P, 216–977–7439.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

Export Administration Regulations (EAR) Notice

This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.

Unclassified-Unlimited
Subject Category: 34
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

A computer program (KTK) has been written for the labyrinth seal Design Model which was developed to permit a rapid and comprehensive calculation for the leakage characteristics of conventional straight and stepped configurations. This is the complete listing of the Fortran computer code.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Labyrinth seals; Seal; Source program; Leakage	71
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	